(19)日本国特許庁（ＪＰ）　(12) **公表特許公報**（Ａ）　(11)特許出願公表番号

特表2002－530731

（P2002－530731A）

(43)公表日　平成14年９月17日(2002.9.17)

(51)Int.Cl.⁷		識別記号	ＦＩ			テーマコード＊(参考)
Ｇ０６Ｆ	12/00	５６０	Ｇ０６Ｆ	12/00	５６０Ｃ	５Ｂ０６０
	12/02	５８０		12/02	５８０Ｊ	
	12/06	５５０		12/06	５５０Ａ	

審査請求　未請求　　予備審査請求　有　　（全 95 頁）

(21)出願番号	特願2000－582866(P2000－582866)
(86)(22)出願日	平成11年11月15日(1999.11.15)
(85)翻訳文提出日	平成13年５月16日(2001.5.16)
(86)国際出願番号	ＰＣＴ／ＵＳ９９／２７０１６
(87)国際公開番号	ＷＯ００／２９９２１
(87)国際公開日	平成12年５月25日(2000.5.25)
(31)優先権主張番号	６０／１０８，９３０
(32)優先日	平成10年11月16日(1998.11.16)
(33)優先権主張国	米国（ＵＳ）
(31)優先権主張番号	０９／４３９，５９１
(32)優先日	平成11年11月12日(1999.11.12)
(33)優先権主張国	米国（ＵＳ）
(81)指定国	ＣＮ，ＤＥ，ＧＢ，ＪＰ，ＫＲ，ＵＳ

(71)出願人　インフィネオン・テクノロジーズ・アーゲー
ＩＮＦＩＮＥＯＮ　ＴＥＣＨＮＯＬＯＧＩＥＳ　ＡＧ
ドイツ連邦共和国　ディー－81541　ミュンヘン，ザンクト・マルティン・シュトラーセ，76

(72)発明者　ストラコブスキー・ヘンリー
アメリカ合衆国　カリフォルニア州95118　サン・ホセ，チェリー・アベニュー，3962

(74)代理人　特許業務法人　明成国際特許事務所

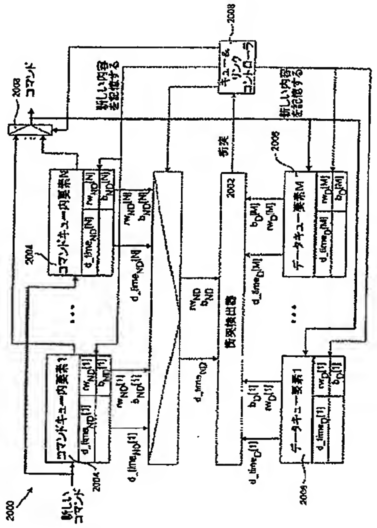
最終頁に続く

(54)【発明の名称】　異常メモリアクセスまたは異なる時間のメモリアクセス実行の際のデータバス上のデータ衝突を検出するための方法および装置

(57)【要約】

【課題】　本発明に従って、ターゲット応答制限に基づいて最適なコマンドシーケンスを実現するためのコマンド再順序付けシステムを開示する。

【解決手段】　コマンドキューに接続されたデータキューは、バーストビットとwビットとを格納するために設けられているとともに、ターゲットデバイスに向けて発行済みの要求に対し、データ転送が制御部との間のデータバス上で現れる時を示す時間を格納するために設けらている。また、システムは、コマンドキューに格納された発行予定のコマンドとデータキューに格納された発行済みのコマンドとの間の、データバス上で起こり得る衝突を検出するために設けられ、データキューおよびコマンドキューに接続された衝突検出器を備えている。キュー＆リンク制御部は、衝突検出器とデータキューとコマンドキューとに接続されており、発行予定のコマンドを格納し再順序付けするために設けられている。ここで、制御部は、データがデータバスに現れる時間とともに、コマンドの新たな発行時間を計算する。



JP Publication 2002-530731

[Detailed Description of the Invention]
[0001]
[Field of the Invention]
Generally this invention relates to a computer system. More specifically, this invention relates to providing access to the shared resource in computer systems, such as a multiprocessor computer system. It is the art for detecting the collision of the data on a data bus especially in the case of unusual memory access or memory access execution of time to differ.
[0002]
[Description of the Prior Art]
In a fundamental computer system, a central processing unit or the CPU operates according to the predetermined program or a set of commands memorized in the related memory. In order to make easy manipulation of information of the central processor under processing besides a set of commands memorized or the program which specifies such operation of a processor, it has memory space in the processor memory or the additional related memory. An additional memory provides the memory location of the information created by the processor, in addition provides the memory location of the information which a processor uses temporarily, i.e., a "memo pad" substitute, carrying out when processing a program. A related memory provides the place in which the print-out of the processor which is performing a set of commands is installed, and the output unit of a system enables it to use the information.
[0003]
In order to access an existing memory, in the system by which much component part (a processor, a hard drive, etc.) must carry out joint use of the one common bus, a possibility that the collision involving access to a memory will occur goes up. Since the system which uses a different processor especially in the case of a multiprocessor computer system etc. is operated simultaneously, access to a memory or other shared resources becomes complicated. Since a possibility that each processor or a processor system will require access to the same memory simultaneously is high, the collision by interprocessor is generally nonavoidable. Fundamentally, operation of two or more processors in a multiprocessor computer system or a processor system produces the intermittent duplication of a memory command to a shared memory or other shared resources as a result.
[0004]
By a certain case, as for the conventional policy taken for solution of the problem of the collision of the memory access demand to a shared memory, perfect duplication of the memory used for each of a processor and isolation of a processor system are included. However, this policy for solving the collision problem of a memory access demand will often make nothing the advantage meant in the multiple processor system. While one processor assists operation of the processor of another side, such a multiprocessor is the most effective when operated in the situation of performing parallel calculation operation with the same data. Conventionally, such a processor system is time common use that processors vie in access to shared resources, such as a memory, or has a port where a processor system is double.

When each processor has a memory bus individually, for example and one side is permitted access, another side is in the state where it is standing by.

[0005]
Various policies have been taken in order to avoid the above-mentioned collision problem. Evasion of a collision is attained by successive use of each processor, or time common use of a processor in a certain policy. In this method, simply, in order to avoid a collision, a processor performs access to a shared resource in order. According to the predetermined continuous action to which such a system generally used is similar with transfer of the ring in a user group by that cause including "ring passing shot" or a "token system", the processor which may collide is polled by a system.

[0006]
Though regrettable, the methodology of successive access by a processor will impose remarkable restriction on overall operation of a computer system. This restriction originates in the fact of spending time when a system is the most in order to poll the processor which collides. In the independent processor's operating, for example, requiring access to a shared memory, delay arises for every memory cycle between processor accesses to a shared resource with execution of the continuous action by a system.

[0007]
Other general policies for collision avoidance are based on the prioritization in the interprocessor within a computer system. In such a method, the priority which followed the hierarchy of the importance of a system to each processor is attached. The processor in which a memory controller has a simple more high priority whenever a collision occurs is provided with access. For example, in the case of the system which has two processors, the first processor and the second processor access to a shared memory. A shared memory is a dynamic DRAM (DRAM) type storage device etc. with which the periodic refreshment of data by which maintenance memory is carried out is needed for a memory typically. Generally, refreshment is performed by another independent refreshment system by a DRAM type memory. In the case of such a multiprocessor system, the both sides of a processor and a refreshment system will compete for access to a shared memory, According to the priority assigned to the processor and the refreshment system, a system memory controller processes a collision and command of a memory access demand. Such a system solves the problem of a collision, and while it is more effective than the system of the collision avoidance by still simpler successive access, it still lacks it in pliability.

[0008]
Other former policies for collision dissolution are the decision making functions incorporated in the memory controller. Though regrettable, since the decision-making portion of a memory controller is operated by the control and timing by a clock system, decision-making is actually performed, and much time will be spent by the time a memory controller can permit access to a shared memory now.

[0009]
Regrettably, the problem of execution of this actual decision-making reduces on parenchyma the access permit capability to the multi bank type memory system which the conventional memory controller has. In a multi bank type memory system, a actual memory core is classified into a specific field, i.e., a bank, and the data read is also

memorized there. While quicker and more efficient memory access can be provided, in order to correspond to a multi bank memory unit, the conventional memory controller will be asked for complicated structure, and, as a result, on the whole, an access speed will fall remarkably as an overall system.

[0010]

As mentioned above, it is clear that the art for detecting the collision of the data on a data bus in the case of unusual memory access or memory access execution of time to differ is demanded.

[0011]

[Summary of Invention]

According to this invention, the art for detecting the collision of the data on a data bus in the case of unusual memory access or memory access execution of time to differ is explained. By forbidding issue of the command which causes a data conflict, The command re-ordering system for realizing the optimal command sequence based on the target response restrictions which determine the optimal slot for the data transfer between an initialization control part and a target subsystem, The command which is due to be published, and time to show the time of data transfer appearing on the data bus between said control section and said target device after said command is published towards said target, It has the command queue formed since the burst bit which shows the burst transmission of data, and a read/write bit (r/w) were stored. The data queue connected to said command queue, Since time to show the time of data transfer appearing on said data bus between said control sections to an issued demand towards said target device while being provided, since said burst bit and said w bit are stored is stored, it provides, and it is ********. Between the command whose system is said issue schedule stored in said command queue, and the issued commands stored in said data queue, It was provided in order to detect the collision which may take place on said data bus, and it has the collision-detection machine connected to said data queue and said command queue. The system was formed in order to store and re-set in order the command which is due to be published, and it is provided with the cue & link control part connected to said collision-detection machine, said data queue, and the command queue.

Said control section calculates the new issue time of a command with the time when data appears in said data bus.

[0012]

About the special feature and advantage of this invention, the much more understanding is attained with the portion and Drawings which are shown in the following of an in [this Description].

[0013]

[Embodiment of the Invention]

Various policies have been taken in order to avoid the collision generated typically, when one or more devices require access to a shared resource in the case of the system by which two or more devices, such as a processor, share the same resource. In a certain policy, the collision is avoided in operating a processor in order, or time common use of a processor attains collision avoidance. In this method, in order to avoid a collision, a processor performs access to a shared resource in order simply. According to the predetermined continuous action to which such a system generally used is similar with transfer of the ring in a user group by that cause including "ring passing shot" or a "token system", the processor which may collide is polled by a system.

[0014]

Though regrettable, in order to spend time most in order that a system may poll the processor which collides, in the methodology of successive access by such a processor, remarkable restriction will be imposed on overall operation of a computer system.

[0015]

Other general policies for collision avoidance are based on the prioritization in the interprocessor within a computer system. In such a method, the priority which followed the hierarchy of the importance of a system to each processor is attached. Such a system solves the problem of a collision, and while it is more effective than the system of the collision avoidance by still simpler successive access, it still lacks it in pliability.

[0016]

Another general policy for collision avoidance contains the decision-making logic included in controller molding equipment. Much time will be spent, by the time actual decision-making is performed from the complexity of decision-making logic and the access permit to the shared memory by a controller becomes possible, though regrettable.

[0017]

This problem on which complicated logic reduces the working speed of a system is still more remarkable in the multi chip module type memory system containing the memory with which it is dotted among two or more storage devices connected mutually, having an operating characteristic which is different in each. the conventional logic compensates the access characteristic peculiar to various storage devices which is alike, respectively and is different -- as -- since it must have been constituted, it is going to solve by reducing the execution performance of an overall system.

[0018]

This invention can be shown in a broad sense as the system 100 by which each has the demand device 102 combined with the universal device controller 104 as shown in drawing 1 A. Here, the demand device 102 is combined with the universal device controller 104 by the system bath 106 appropriately formed so that access to the shared resource 108 of arbitrary numbers and a kind might be provided. The system bath 106 is combined with the universal controller 104 by the related system interface layer 110 in one embodiment, and further, The universal controller 104 is combined with the shared resource 108 by the shared resource interface 109. The universal controller 104 is set to a broad sense based on the demand to the shared resource from the arbitrary demand systems 102, and the shared resource operating characteristic parameter 113 judge the state of the shared resource 108.

[0019]

The demand system 102 is one processor in a multiprocessor system, When access to the shared resource 108 as the storage device 108 currently shared by other processors similarly combined there is required, The universal controller 104 determines an order of the operation which should be performed, in order to complete a desired resource access request. For example, when the storage device 108 is SDRAM, the operation contains precharge, page closing, page opening and a page lead, or a page light typically.

[0020]

If a certain specific operation order is determined, in order to avoid a data collision or confrontation of other kinds, for example, the suitable time interval between the continuous action which the universal controller 104 was able to set in order is determined. According to a desirable embodiment, the time interval is determined selectively based on the operating characteristic of the shared memory device memorized by the look-up table, for example. Then, the access command which was able to be appropriately set in order is outputted by a universal controller, ranks second, and is answered by the shared memory.

[0021]

By the detailed description of the invention shown below, in order to promote a perfect understanding about an invention, two or more concrete embodiments are shown. However, in this invention, it does not depend on these specific details, or other elements or processes are used so that it may become clear to a person skilled in the art. Therefore, it may practice.

In other case, in order to avoid that the essence of this invention becomes ambiguous, a well-known process, a procedure, a component, and the detailed explanation about a circuit are omitted.

[0022]

Hereafter, this invention is explained from a viewpoint of the memory controller set up function as a communicator between a processor and a shared memory. However, this invention should not be concerned with whether it is common use, but should care about that it can perform also as a universal controller which can control access to arbitrary resources. There is no necessity that such a resource is a memory, and the waiting time of bus access is reduced in the fact and this invention, for example.

Therefore, it may be used also in order to control access to a common system bath, such as controlling the amount of information in a multiprocessor for the purpose of making the efficient bandwidth of a system bath increase.

[0023]

Next, by drawing 1 B, it has the demand devices 102, such as a processor, and the system 100 combined with the universal controller 104 by the system bath 106 is shown. The controller 104 is combined with the shared resource 108, and the shared resource 108 is the memory 108 which can take various gestalten, for example.

They are DRAM, SDRAM, SLDRAM, EDO, FPM, or RDRAM.

According to the embodiment of illustration, the system bath 106 transmits the memory address demand to which the one-way address bus 106-1 is outputted by the processor 102 to the universal controller 104 including the one-way address bus 106-1. Furthermore, also including the one-way address bus 106-2, the one-way address bus 106-2 coordinates the system bath 106 with the address bus 106-1, and it transmits the command about a memory address. For example, when the excecutable command in which the processor 102 is memorized by the specific memory location in the memory 108 is required, A processor outputs a read request (a system command is called) to the command bus 106-2, and the output of a memory address demand (a system address is called) corresponding also to the address bus 106-1 is performed simultaneously substantially. The both sides of a system address and a system command are received by the system interface 110 which is contained in the controller 104 and which can be set up. The thing of any methods as which the system command and system address which were thought that setting out is possible here are required by the memory 108, and a gestalt so that the system interface 110 can process it, It means that the system interface 110 can be set up. The necessity that the processor 102 emits an individual demand to each storage device is lost by this, and the data which the processor 102 needs can be stored in the arbitrary numbers combined with the controller 104, and the storage device of a kind.

[0024]

The system interface 110 is set up in the embodiment of illustration change the system command and system address which were received into what is called the universal command 200. The example of the universal command 200 is shown in drawing 2 A. When a shared resource is a DRAM type storage device (SLDRAM, SDRAM, EDO DRAM, etc. are included), in one working example the universal command 200, It is formed from five data fields including all the operations required in order to perform the arbitrary memory access demands of the memory 108. The precharge operation used in order to indicate whether this field has the necessity of charging the specific low beforehand, including the precharge operation such operation is indicated to be by the data precharge field 202 is included. As other operations, the data activation field 204, the data read field 206, the data write field 208, and datary Freshfield 210 are included. For example, the memory 108 is active in the memory bank 1 here now (.). That is, it has the memory page 1 opened after the lead or the light was performed, and the continuing processor command reads the data memorized on the page 2 of the memory bank 1, and assumes that it is demanding to output to the processor 102. In this case, in order to execute the command demanded by the processor 102, the page 1 must be closed (that is, precharge of the page 1 is carried out), and the page 2 must be activated. After activation is completed, the lead from the page 2 is performed. Therefore, the universal command 212 shown in drawing 2 B, It is generated by the universal command generation part 110 which has the data fields 202 and 204,206,208,210, The data field 202,204,206 of them is set as "1" which shows "execution of related operation", and the data field 208,210 is set as "0" which shows "the nonfulfilment of related operation" (namely, "NOP").

[0025]

It returns to drawing 1 B. Since access to the memory 108 is shared by several different demand devices, it is dramatically dynamic, therefore the state of the memory 108 always changes. The state of a memory means that it is necessary to get to know the state of the memory location, in order to perform specific operation rightly in a specific memory location. For example, when the specific memory page closes, in order to perform read operation, it is necessary to open the memory page first. Therefore, in order to detect the state in the time of the specific address position, the newest operation performed to the specific memory location is identified with the resource tag 300 shown in drawing 3. In one embodiment of this invention, the resource tag 300, In order to identify the command published at the end to the address field 302 used in order to identify a specific memory address position, and the address identified by the address field 302. The final release command field 304 used and also the final command issue time data field 306 are included. For example, the resource tag 308 of memory address ADD_5 shows that the page read command was published at the time 5 phi (5 system clock cycles are shown), The resource tag 310 shows that a page light is performed by the memory page at the time 10 phi about the same memory address ADD_5. By observing the state of memory address ADD_5, the universal controller 104 recognizes that the memory page in ADD_5

has already opened, therefore that page open operation is unnecessary.

[0026]

The information about the state of the resource provided with the tag 300 memorized by the resource tag buffer 114 to origin. The command ordering machine 116 combined with the system interface 110 which can be set up, Show the suitable time interval between each of the command components 202-210 of the universal command 200, and by that cause, The command 220 constituted so that the command components 202-204 as shown in drawing 2 C, and the command components 204-206 might become time interval t_1 and t_2, respectively set in order is provided. Since the command components 208-210 are the "NOP" type fields here, The command 220 set in order is needing an equal time period for the sum total of an equal time period and ** substantially at a clock period required for the components 202-206, and t_1+t_2, also excluding what kind of reference about these fields. Thereby, the command ordering machine 116 can provide the command between the processor 102 and the memory 108, and the optimal flow of data.

[0027]

When the shared resources 108 are multi bank type storage devices, such as SDRAM, in another embodiment of an invention, Or when shared resources are multi-equipment memories, such as a multi chip module, the resource tag buffer 114 can memorize the resource tag about all the pages opened, for example in a specific bank or device. In one working example, a comparator (not shown) detects the bank number or device identifier in a system address, and compares a page address and a system address with the contents of the tag buffer 114. When a comparison result is not "coincidence" (namely, when an address does not suit), The universal controller 104 must close an old page using the address from the tag buffer 114, and must open a new page based on a still newer system command.

[0028]

When several different devices are assisted by the universal controller 104, it is the operation parameters only relevant to a specific device, and it is desirable that the operation parameters also in relation to the system address inputted can be chosen. When the universal controller has assisted several different devices, the address space controller 120 combined with the universal controller 104 is shown in drawing 1 C. According to the embodiment of illustration, the address space controller 120 has the performance which chooses only the parameter peculiar to a device which shows one device relevant to the inputted system address. The system address into which, as for the address space controller 120, the comparator 122 was inputted in concrete working example shown in drawing 1 D including the comparator 122, The contents of the regional-address range buffer 124 which identifies the device (or making it the same memory area) relevant to the inputted address are compared. Discernment of a specific device or field will choose one register among the group of the device parameter registers 126 and 128 (it is combined with the buffer 124 and each contains a characteristic parameter for every specific device). Selected device parameter registers rank second and present the specific operation parameters about the device corresponding to the system address. According to another embodiment, the contents of selected device parameter registers are inputted into LUT118. The device with which arbitrary numbers differ by this can be assisted by the universal controller 104, the specific operation parameters of each device are identified, and it is used for the optimal ordering of a corresponding universal command.

[0029]

when one of the devices combined with the universal controller says a new command that there are no reception ** by a busy state, the point that it is useful that another waiting arbitrary commands can be chosen by series of commands should be cared about. In another embodiment of an invention, as each of the response by a device and the demand by a universal controller has the identification number 150 related, respectively and it is shown to drawing 1 E by the embodiment of illustration, the identification number is the data word of 5 bit length. The identification number 150 is formed so that the group selector field 152 of 2 bit length and the demand number field 153 of triplet length may be included. A group selector (GS) judges to which group a specific system demand belongs (for example, processor), and a demand number (RN) expresses an affiliate group's demand or the number of a response identified by the group selector field 152. In that case, the demand which continues from the same transceiver has the continuous demand number.

[0030]

According to another embodiment, the answering group or demand group who has a higher priority value precedes the group priority selector register 154 with the group of a low priority value including the priority value about each of an answering group or a demand group. When a low demand or response of a priority value must have been processed with the following clock period by this, it precedes with a demand or response of the low priority value, and a demand or response of a value may be processed at least for Takayoshi. In order to prevent what is called a ** lock (a live lock, Livelock), the ** lock counter register 156 includes a continuous number of a demand (or response) which can be preceded with the demand (or response) of a low priority value of information which have a high priority value. By this, the demand (or response) of a low priority value needs to be neglected among the cycle of many of clock periods.

[0031]

In order to optimize control of the both sides of the flow of a command, and data flow as another important matter, the point that each of a shared resource is relating a set of operating characteristics (it is access time, CAS waiting time, etc. in the case of for example, DRAM molding equipment) to each is got. When one or more shared resources are assisted by the universal controller 104, each of a shared resource has a set of different operating characteristics, and in another embodiment. The operating characteristic is memorized by the look-up table (LUT) 118 combined with the command ordering machine 116. Using the information which is coordinated with the resource tag memorized by the resource tag buffer 114, and LUT118 provides, the command ordering machine 116 sets in order the command components 202-210 appropriately, and forms the command 220 set in order. This is applied especially when shared resources are sets of the storage device with which each just has a substantially different operating characteristic, such as a multi chip module.

[0032]

Next, drawing 4 is the flow chart with which the universal controller showed the details of the process 400 of aiming at access to a shared resource, according to the embodiment of an invention. A system starts this process 400 from 402 which requires access to a shared resource. When a shared resource is a DRAM type storage device, the operation contains precharge, refreshment, closing, opening, a lead, and a light. For example, when a processor generates the system address of the relation which shows the place in the memory the page demanded as the system command (namely, page lead) is remembered to be, the memory page memorized in a shared memory is required. In an embodiment with this preferred, the state of a resource is judged in 404 using the resource tag relevant to the active memory location within a shared memory. Subsequently, the judgment about ordering [required of 406 in order to perform the demand of the request which carries out a shared resource pair] of operation is performed. In 408, a universal controller generates a universal command based on a continuation order of operation required in order to perform a desired demand. For example, in order to perform page read operation, the page opened on the preceding paragraph story is closed, and a new page is activated, and read operation needs to be performed. In that case, these [all] are understood with one universal command composition. If a universal controller forms a universal command using the resource tag about a shared resource, and a characteristic operating characteristic, it will rank second and a universal controller will judge the suitable time interval between each command component of a universal command in 410. Then, in 412, the command set in order is published to a shared resource. Under the present circumstances, a physical stage is used in another embodiment. Finally in 414, a shared resource answers the command set in order by showing the data memorized by the position shown in a system address, for example etc.

[0033]

According to another embodiment of an invention, a universal controller determines the state (404) of a resource, and the turn (406) of operation to perform using the process 500 shown in drawing 5. This process 500 is started by comparing a resource Type identifier (namely, memory address register) with a resource identifier (namely, resource tag address field 202) by 502. When generating of "coincidence" is checked in 504 (namely, when the address of a new command suits with the tag address field in the time), the following command is published in 506. On the other hand, when a new command address does not suit the tag address field in the time, it ranks second (namely, when inharmonious) and the judgment of whether the page old at 508 still opens is performed. When the old page opens, the page is closed in 510 and it is opened by the page new at 512. However, when it is checked that the page old at 508 does not open, it ranks second and is opened by the new page in 512. If opened by the once new page in both case, the following command (data manipulation) will be published in 506.

[0034]

In another embodiment of an invention, a universal controller is based on the process 600 shown in drawing 6, and determines the suitable time interval between each of continuous action (410). The command of the beginning [on 602 and as opposed to / process / 600 / this / a certain specific resource in a universal controller] of a series of new commands, It is started by comparing the command of the last of a series of newest commands among the commands published to the same resource by the point in time. In 604, a universal controller judges the time restrictions between universal command components by comparing the final command element of the newest command with the new command component of the beginning of a universal command among the things before it. According to another embodiment, a universal controller uses 2 index look-up table (LUT) which takes the gestalt of two-dimensional array as shown in Table 1. In it, the command with the first old (that is, the newest among former) line of arrangement is expressed, and the first row expresses the new command of arrangement. For example, when an old command is a page lead with reference to Table 1, And when a new command is a page closing, the minimum time quantity (namely, physical minimum time which issue takes) permitted between these two operations is shown in the intersecting position of the new command of those page closing, and the old command of a page lead. Usually, the information memorized by LUT is provided by the manufacturer of a shared resource.

[0035]

[Table 1]

表 1

		古いコマンド			
		ページクローズ	ページオープン	リード	ライト
新しいコマンド	ページクローズ			5φ	
	ページオープン				
	リード				
	ライト				

[0036]

In 606, a judgment of physical restrictions of the resource about a specific universal command component will perform the judgment of whether the further command component exists in the same universal command. When the further command component does not exist, in 608, a universal command and the details about the related time interval of a component are memorized. On the other hand, when the further command component is included in the universal command, control is returned to 604 and the physical temporal restriction which corresponds about the component is judged.

[0037]

However, in order to observe the state of the physical page in the shared memory 108 which has two or more memory banks, for example, very many resource tags are needed and very much cache memory is needed according to it for the resource tag buffer 114. In order that each may search with this the peculiar resource tag about the specific memory page of a memory located in remoteness, very much time will be required, and as a result, the overall working speed of the universal controller 104 is made reduced. According to another embodiment shown in drawing 7 A, a page hit / beauty pageant troller 702 is contained in the universal controller 104, and it is set up so that severalM of the page register 704 may become less than the number N of memory banks in the multi bank memory 706. This is because it is not a reason although all the banks can be responded in the M page registers 704. In this operation, each of the M page register 704 memorizes the address and condition data of an open ****** page. The random page register number generation part 708 generates the integral value below random M corresponding to a page register, and exchanges it for the address of the opened page. The comparator 710 performs comparison with the inputted system address, and the bank number of all the M registers and a page address in parallel, and acquires either of four possible results shown below.

[0038]

1) When the comparator 710 shows a hit (coincidence), it is opened by the page of a request of the bank demanded and the preparation for access is completed.

[0039]

2) When the comparator 710 shows the mistake (disagreement) which is the hit (coincidence) and page of a bank, the universal controller 104, It is necessary to close an old page using the page address from a page register, and to open a new page using the page address from a system address.

[0040]

3) When the comparator 710 shows a mistake in the both sides which are a bank and a page, the universal controller 104, It is necessary to close the old arbitrary pages in the bank of the number provided by the random page register number generation part, and to open a new page using a system address. Then, access to a desired bank is performed.

[0041]

4) The both sides of a bank and a page are mistakes, however when at least one page register is intact, the register is used and it is opened by the new page.

[0042]

As shown in drawing 7 B, another embodiment is substituted for the random page register number generation part 708 by the comparator (LRU) 712 used most before, and it judges which [of the M registers 704] is intact for a long time by it (that is, were used before most?).

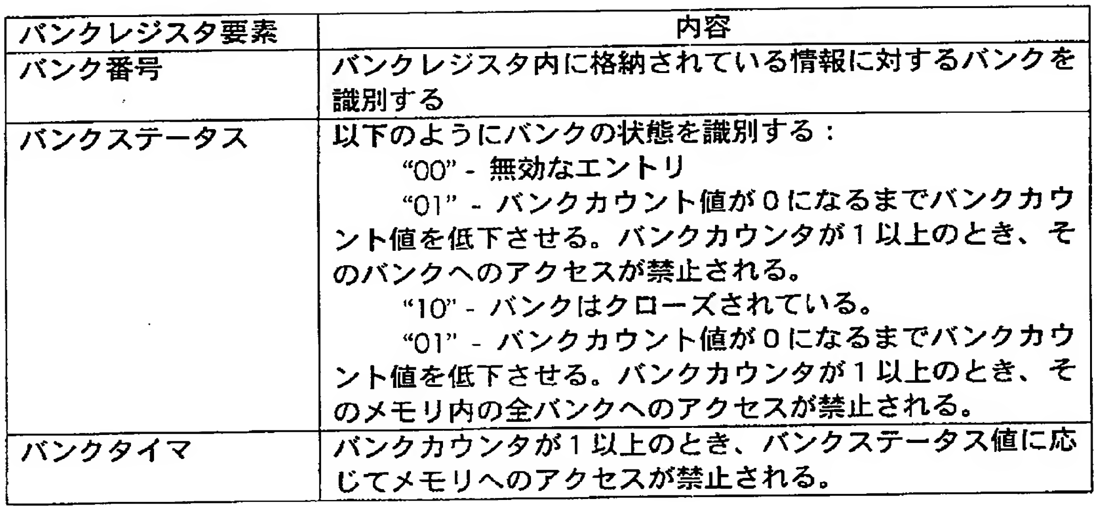
[0043]

In addition to observation of the state of the physical page in the multi bank memory 706, the bank access controller 800 shown in drawing 8 contains the N bank resisters 802 corresponding to the memory bank N [several] contained in the multi bank memory 706. The bank resister 802 the information on a related bank is remembered to be includes the bank number field 804 which specifies the identification number of a bank. The bank resister 802 includes the bank status fields 806 which show the state of the specific bank identified with the bank number in the bank number field 804. According to a concrete embodiment, the bank status fields 806 can take a value as shown in Table 2.

[0044]

[Table 2]

表　2



バンクレジスタ要素	内容
バンク番号	バンクレジスタ内に格納されている情報に対するバンクを識別する
バンクステータス	以下のようにバンクの状態を識別する： “00” - 無効なエントリ “01” - バンクカウント値が0になるまでバンクカウント値を低下させる。バンクカウンタが1以上のとき、そのバンクへのアクセスが禁止される。 “10” - バンクはクローズされている。 “01” - バンクカウント値が0になるまでバンクカウント値を低下させる。バンクカウンタが1以上のとき、そのメモリ内の全バンクへのアクセスが禁止される。
バンクタイマ	バンクカウンタが1以上のとき、バンクステータス値に応じてメモリへのアクセスが禁止される。

[0045]
The problem resulting from the collision of memory access is increasing increasingly with development of packet inclination type high speed memories, such as synchronous link DRAM (SLDRAM) which transmits bus data at the rate of the range of 400 - 800 Mb/s / pin. When drawing 9 A is referred to first, the illustration SLDRAM type multiprocessor system 900 by the embodiment of an invention is shown. The multiprocessor system 900 contains the processor 902 connected to the controller 904 by the system bath 906. The universal controller 904 ranks second and is connected to the synchronous links DRAM (SLDRAM)908 and SLDRAM910 by SLDRAM bus. The SLDRAM bus comprises the one-way command bus 912 and the bidirectional data bus 914. Although only two SLDRAMs are shown in drawing 9 A as an important matter, it is raised by bus 912 and 914 that arbitrary numbers of SLDRAMs may be connected to the universal controller 904. In the case of being another, SLDRAM can take the gestalt of the module with a buffer in which only arbitrary suitable numbers contain a SLDRAM like SLDRAM908. Initialization/synchronous (I/S) bus 916 which connects the universal controller 904 to each of SLDRAM908 and 910 provides the signal path of the initializing signal generated by the universal controller 904, and a synchronized signal.

[0046]
According to another embodiment of an invention, the command bus 912 top is selectively transmitted to the command, the address, and control information on the universal controller 904 which were packet-ized by SLDRAM908 and 910. The data bus 914 is set up transmit the right data packet-ized from the universal controller 904 to those whom SLDRAM908 and 910 chose either. Or the data bus 914 is formed also so that the lead data packet-ized from the direction where SLDRAM908 and 910 were chosen either may be returned to the universal controller 904. As an important matter, it is raised that the command bus 912 and the data bus 914 usually operate at the same speed mutually [for example, 400 MB/s/p 600 MB/s/p, 800 MB/s/p, etc.].

[0047]
Two or more control signals which are generated by the universal controller 904 and transmitted by the command bus 912 include the run clock signal (CCLK), the FLAG signal, command address signal CA, the LISTEN signal, LINKON signal, and RESET signal of no difference, for example. Usually, a packet command comprises four 10 continuous bit word, and, as for the word of the beginning of a command, the bit of the beginning of a FLAG signal has become "1." According to a desirable embodiment, since a command word is latched, the both ends of run clock signal CCLK of no difference are used by SLDRAM908 and 910. SLDRAM908 and 910 answer the LISTEN signal of H level by investigating the command bus 912 about the inputted command. Or SLDRAM908 and 910 answer the LISTEN signal of L level by going into a power-saving standby mode. or [that a LINKON signal and a RESET signal stop the direction where SLDRAM908 and 910 were chosen either so that it may be in the state of a known request] -- or in order to start, it is used, respectively.

[0048]
It will be discussed only after it is understood well that arbitrary numbers of SLDRAMs considered that SLDRAM908 is suitable may be connected to a universal controller about the point that the argument was left behind here. A typical SLDRAM device like SLDRAM908 is composed hierarchical not only by a memory area but by a memory bank, a column, a low, and a bit as discussed upwards. It should be cautious of it actually being accepted that each of these layer levels have an operating characteristic which is mutually different. Although parameters, such as memory access time, chip enable time, and data retrieval time, are included in such an operating characteristic, limitation is not carried out to these. It should comment on the bank in a multi bank memory having the usually same operating characteristic to a field being defined as a different device called a different memory type and memory group with command waiting time different, respectively or data waiting time. For example, it is connectable with the second un-local memory group located on the board to which the driver who one local memory group can connect with a memory controller direct, and mediates the command waiting time and data waiting time about a local memory group makes it increase. In other cases, it is thought that each of various memory chips which form a multi chip module is a different memory area.

[0049]
When the system of drawing 9 A is described still more concretely, SLDRAM908 is a multi chip module which has the four accessible memory chips A, B, and C and D individually by the command bus 912, the data bus 914, and I/S bus 916, respectively. In order that each of the memory chips A-D may carry out scheduling of a command or the data packet the optimal, It can have a different operating characteristic (usually given by the manufacturer), and the universal controller 904 can use a specific layer level and/or the corresponding operating characteristic of a memory area.

[0050]
As an example, drawing 9 B shows the typical timing diagram of the typical SLDRAM bus transactions according to the multiprocessor system shown in drawing 9. A processor usually generates during an operation the processor command packet which the suitable memory bank (plurality is also good) of SLDRAM908 answers like the read command 950 or the write command 952. Usually, the read command 950 and the write command

952 are pipelined on the system bath 906 based on the specific demand of the processor 902 by which they are generated.
It is not suitable for the optimal performance of SLDRAM.
System clock CLK_{sys} (not shown) gives a required timing signal.
[0051]
In this example, the processor 902a generates the read command 950 with memory address MA_1 located in the memory chip A of SLDRAM908. On the other hand, the processor 902b generates the read command 952 with memory address MA_2 similarly located in the memory chip A of SLDRAM908. In this example, the read command 950 is an output to the system bus 906 to which priority is given over the output of the write command 952. The universal controller 904 begins to process a command using the destination address specific information which received the read command 950 first, and was continuously stored in the universal controller 904 based on the command itself and command address MA_1. Once it decides on the shortest issue time, next, according to the received processor command 950, the universal controller 904 will generate the SLDRAM command packet lead 960, and will send it out to the command bus 912.
[0052]
Generally, the SLDRAM command packet is composed as four 10 bit word showing SLDRAM of 64M with eight banks, 1024 row addresses, and 128 column addresses, as shown in Table 3. As shown, in a triplet and a row address (ROW), 10 bits and the column address (COL) of a bank address (BNK) are 7 bits. Otherwise organization and densities of many are possible, and it is accommodated in any formats of the others by which it is specified that it is suitable and in which it deals besides the illustrated 40-bit format, and gets. While the power supply is on, the universal controller 904 composes a command packet based on polling of SLDRAM about factors, such as a number of a bank, a low, and a column, and an associated operating characteristic which are then stored by the universal controller 904.
[0053]
The word of the beginning of a command packet contains two or more chip bit IDs. Which disregards the command which does not adjust SLDRAM to local ID. By the universal controller 904, chip ID is assigned using an initializing signal and a synchronized signal with a power turn. Thus, the universal controller 904 is addressed by a one to one correspondence to each SLDRAM of the multiprocessor system 900 by generation of a separate chip enable signal or glue logic.
[0054]
[Table 3]

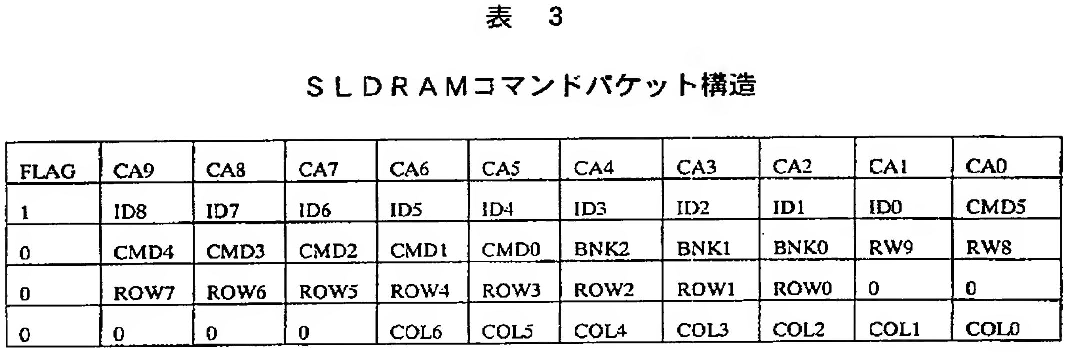


表 3

ＳＬＤＲＡＭコマンドパケット構造

FLAG	CA9	CA8	CA7	CA6	CA5	CA4	CA3	CA2	CA1	CA0
1	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0	CMD5
0	CMD4	CMD3	CMD2	CMD1	CMD0	BNK2	BNK1	BNK0	RW9	RW8
0	ROW7	ROW6	ROW5	ROW4	ROW3	ROW2	ROW1	ROW0	0	0
0	0	0	0	COL6	COL5	COL4	COL3	COL2	COL1	COL0

[0055]
Since the read command 950 and the write command 952 are pipelined, the universal controller 904, After the read command 950 receives, after fixed time, the write command 952 is received (or it can also be made to store in a buffer), and it ranks second, and SLDRAM command packet writing 962 corresponding to the write command 952 is published. Since the same bank (A) is accessed by both commands, in order to avoid interference of the read command 960 published before, the universal controller 904, The issue time (namely, time of issue) of the read command 960 for making the specific characteristic data and the shortest issue time of MA_2 generate and data offset of the light 962 are used.
[0056]
Thus, the universal controller 904 can carry out scheduling of the issue of a SLDRAM command packet dynamically at least based on a command, or the present state of the stream of a data packet and the operating characteristic of a specific destination address device.
[0057]
Next, drawing 10 illustrating the block diagram of the memory controller 1000 according to the embodiment of this invention is mentioned. It should annotate that it should not be realized that the memory controller 1000 is only one of the embodiments of the universal controller 104 shown in drawing 1, therefore the limit of this invention is restricted. As for the memory controller 1000, this connects the processor 902 to the memory scheduler 1006 (it is called a scheduler) via the system bus 906 including the system interface 1002. In one of the embodiments of this invention, SI 1002, It is arranged in order to prepare for transmission to the memory command packet scheduler 1004 of both the memory command packet generated by the processor 902 and the write-in data packet related with it. In the situation which the scheduler 1006 will show if the command in which an internal buffer is full and new cannot be accommodated, SI 1002 holds a new command until the scheduler 1006 accepts a new command and displays it as a preparation completion.
[0058]
The synchronous link media access controller (SLiMAC) 1008 provides the scheduler 1006 and the physical interface between SLDRAM908. SLiMAC1008 still more specifically includes the command interface 1010 and the data interface 1012 which connect SLiMAC1008 to SLDRAM908 via the command bus 912 and the data bus 914, respectively. According to the suitable embodiment of this invention, the command interface 1010 transmits a memory command to SLDRAM908 from SLiMAC100 with the related command clock CCLK. According to some embodiments, in order to make command clock signal CCLK which usually operates at 200 MHz generate, SLiMAC1008 has incorporated the clock doubler using interface clock signal ICLK (it can operate at about 100 MHz).
[0059]
In one of the embodiments of this invention, the data interface 1012 carries out both reception of the data of the data bus 914, and transmission. Being made to sufficient size to support a number of SLDRAMs needed should comment on the width of the data bus

914. Therefore, in order to supply required bandwidth, enough data interfaces may be included in SLiMAC as required. As an example, if the data bus 914 is 32 bit width (for example, 16 bits per SLDRAM), SLiMAC1008 can be provided with two data interfaces which can control 16 bits related with each SLDRAM, respectively in that case. Thus, the size of the data interface included in SLiMAC1008 can be strictly fitted to the specific composition of SLDRAM connected to it.

[0060]

By the almost same method as using the command interface 1010, SLiMAC1008 can supply data clock signal DCLK accompanying the lead data transmitted to SLiMAC1008 from SLDRAM908. In one of the embodiments of this invention, the data clock DCLK is generated by using the clock doubler which makes interface clock ICLK frequency increase from about 100 MHz to about 1000 MHz. It should comment also on interface clock signal ICLK, command clock signal CCLK, and data clock signal DCLK being all the phase simulation-like.

[0061]

In the suitable embodiment of this invention, the scheduler 1006 contains the limit block 1016 arranged so that a system command and the system address information related with it might be received from SI 1002 to which it was connected. The limit block 1016 supplies the timing information related with SLDRAM command packet data to the re-ordering block 1018. The buffering write 1020 receives lead data from SI 1002. As directed by the scheduler 1006, lead data is transmitted from the data interface 1002 through the read buffer 1022 connected to the data bus 914 arranged so that lead data might be supplied to the system interface 1002. The I/S block 1024 connected to initialization/synchronous (I/S) bus 914 supplies suitable initialization and/or synchronized signal to SLDRAM908 as a demand.

[0062]

Setting working, the scheduler 1006 receives the pipelined memory command packet which was generated by the processor 902. Usually, the memory command packet comprises a memory command and a memory address related with it. In one of the embodiments of this invention, the scheduler 1006 decodes the memory address related with the received new command, in order that a memory command and the data packet related with it may determine the destination address dispatched (if it was). Once it is decoded, in order to send out a new SLDRAM command packet, the destination address specified device characteristic data stored there will be used for the scheduler 1006 with the information related with the memory command sent out immediately before. A new SLDRAM command packet is an output to the command bus 912, and is an output to SLDRAM identified by chip ID eventually contained in the SLDRAM command packet.

[0063]

As a part of scheduling process, the scheduler 1006 determines the amount of shortest time after issue of the demanded command which was published immediately before before issue of a new command. As mentioned above, for example, each layer level of SLDRAMs, such as a memory bank, By that (usually given by the manufacturer) which can have a different operating characteristic, the scheduler 1006 is polled to each of SLDRAM which it serves between initialization. According to some embodiments, when not allowing polling for the connected memory device to determine an operating characteristic, memory specific parameters (timing etc.) can be directly written in the limit block register 1016. Once SLDRAM is polled, the scheduler 1006 stores the device specific information used in order to develop a suitable scheduling protocol later. Thus, the scheduler 1006 can supply the scheduling service which the procedure which waste of the further time and cost require does not have, either, and is adapted also for what number and type of SLDRAM, without needing hard wiring.

[0064]

Drawing 11 illustrates typically the limit block 1100 according to the embodiment of this invention. It should comment on the limit block 1100 not being the only possible embodiment of the limit block shown in drawing 10, therefore it not being what is limited to this. The limit block 1100 is arranged in order to decode the new address signal which was related with the new memory command generated by the processor 902 and which received, and it contains the address decoder 1102 connected to SI 1002. The decoded new address signal supplies the input to the arrangement tag register 1104, it is for [of a related SLDRAM memory bank] all the, or the status and other pertinent information only for a subset are stored in this arrangement tag register depending on the case. The arrangement tag register 1104 supplies the input to the selector 1106, and the selector 1106 transmits the data relevant to the selected virtual bank to the look-up table (LUT) 1108 based on the decoded new command.

[0065]

The limit block 1100 contains again the field comparator 1110 connected to SI 1002, and this field comparator 1110 supplies the area identification child who shows the memory area where a new command address exists using the new address signal which received. Thus, it is based on memory area specific characteristic data at least selectively, and the limit block 1100 can provide the best scheduling protocol for a new memory command. The field comparator 1110 supplies an area identification child to LUT 1108 as an input with a new command signal. Next, LUT1108 supplies the minimum delta issue time and data offset which are used in order to change into a SLDRAM command packet the new address related with a new command and it. It should comment on the delta time (with clock cycle) when the minimum delta issue time publishes a new command in a relation with the old command published immediately before being shown. Data offset time expresses the delta time in the clock cycle for receiving the lead data packet related with a command new after issue of a new command.

[0066]

In one embodiment of this invention, the limit block 1100 can store four different parameter sets including 16 arrangement tag bank resisters for each four timing field in which LUT1108 has 16 associated registers.

[0067]

Drawing 12 is the timing diagram 1200 of the SLDRAM bus signal which answers the received processor command according to one of the embodiments of this invention. When Table 4 identifies various generated signals, it should comment on summarizing the scheduling process performed by the limit block. A memory command takes the form of {a command and an address}, a "command" expresses here the command which should be executed, and it should comment also on the "address" expressing the location of the associated memory.

[0068]

Next, Table 4 and drawing 12 are mentioned. Between system clock cycle phi_1, it is received by the address decoder 302 and the first command {open page and 1000} is received by the field comparator 1110 in parallel. In this example, the address decoder 1102 decodes an open page command address "1000" as "100" and "400", and it is determined that the field comparator 1110 is contained in the memory area 0 in it. Since an open page command is the first command that should be received, there is "no hit" also in which of virtual bank B_0-13, and a corresponding substitution counter is set as "0." At this embodiment, random counting or other suitable methods are used by other embodiments to a substitution counter being updated based on the pseudo-random counting method. Since the first command {open page and 1000} is an open type command, there is also no associated minimum delta issue time and data offset, therefore the page of the address 1000 is opened by the first command clock cycle $phiC_1$.

[0069]

Between system clock cycle phi_2 of the following, {lead and 1000} command is received by the limit block 1100, The address decoder 1102 decodes it as 100 and 400 (that is, lead the page opened by the front clock cycle for the memory address location 1000), and these values make the field comparator 1110 set an area identification child as the field 1. However, in this case, by other ways of speaking before being stored in the B_0 register, "an old command" "will hit" in B_0, and, thereby, outputs a selector to LUT1108 by considering "a lead" as an "old command" input. As other inputs, the field sign child "field 1" emitted by the field comparator 1104 and the "new command" input which is leads are included. LUT1108 generates the three minimum delta issue time of command clock cycle phi_3 using the stored characteristic data. This shows that at least three command clock cycles must separate issue of {page opening and 1000} command, and the {lead and 1000} command related with it.

[0070]

Thus, it is processed based on the command which was published immediately before according to the characteristic data in which each memory command packet received in the limit block 1100 was stored in LUT1108 as for a certain grade at least.

[0071]

Next, re-ordering of the command received from the limit block according to the specific embodiment of this invention is explained. Drawing 13 A - drawing 13 C are the timetables 1302-1304.

It is useful to let the example of simple command re-ordering pass, and to illustrate some of advantages realized by re-ordering of a memory command according to the specific embodiment of this invention.

Each timetable shows four read commands corresponding to two different memory banks. CMD0 and CMD1 are the read commands dispatched to the bank 1 of the associated memory. CMD2 and CMD3 are the read commands dispatched to the bank 2 of the associated memory. The timetable 1302 is an order that a command is received by the memory controller from a system processor, and shows the memory command arranged on the command bus which connects a memory with a memory controller. CMD0 occupied the time zone 0, CND1 occupied the time zone 3, CMD2 occupied the time zone 4, and CMD3 occupies the time zone 7. Each time zone expresses one clock cycle.

[0072]

As stated above, in order to process the command published before, the shortest delay is required for the command to the same memory bank between issue. This is expressed to drawing 13 A by two time zones between the commands of a couple. Supposing four read commands are sent to a memory in the order shown in drawing 13 A so that it can grasp, it will pass over a command bus between the four usable clock cycles 1, 2, 5, and 6, i.e., time zones, without being used. Re-ordering of this inefficient command to which or followed this invention a little at least will improve so that it may discuss below.

[0073]

With re-ordering of the command of drawing 13 A which followed the specific embodiment of this invention, respectively, even if the timetables 1304 and 1306 of drawing 13 B and drawing 13 C have few advantages acquired by that cause, they are illustrating some. In this example, the conflict of a data bus expects **** and is not taken into consideration. However, for effective re-ordering of a memory command, attention must be paid to such consideration so that it may discuss below. For the fact that CMD2 and CMD3 are dispatched to a memory bank which is different in CMD0 and CMD1, the memory access waiting time among two pairs of commands may not pose a problem, but may be disregarded. That is, a command can be rearranged as shown in the timetable 1304, it places CMD2 in the time zone 1 just behind CMD0, and places CMD3 in the time zone 4 just behind CMD1. It is because delay is unnecessary by the fact that this is dispatched to a different memory bank, between issue of CMD1 and CMD3 during issue of CMD0 and CMD2. However, between the pairs of the command dispatched to the same bank, it will be understood that the shortest time delay, for example, two clock cycles, must be maintained, as shown in drawing 13 C. That is, the trial which cuts down the time delay between the continuous commands to the memory bank with the same re-ordering of a command is not included.

[0074]

The result of re-ordering of a command is shown in drawing 13 C. Here, four commands were published in 5 clock cycles, and it has passed, without using only the time zone 2. Of course, the 5th memory command that goes to another memory bank will be inserted in the time zone 2, and increasing further the efficiency for which a command bus is utilized to a limit will be understood.

[0075]

Drawing 14 is some block diagrams of the memory controller constituted according to the specific embodiment of this invention. The re-ordering circuit 1400 receives the memory command sequence which enters from a system processor, i.e., a command sequence called 1, 2, and 3. According to the specific embodiment, a memory command is transmitted to the re-ordering circuit 1400 via a limiting circuit (not shown), and a limiting circuit as mentioned above, Restrictions of issue time are imposed according to other commands dispatched to the logic bank with same memory related with the selected command. A command is re-set in order within the command queue 1402, and a command is published towards a memory from there. In this example, the command is re-set in order in order of 1, 3, and 2.

[0076]

An original memory command sequence, i.e., 1, 2, and 3, is stored in the FIFO memory in the data reading circuit 1406. The sequence in FIFO1404 is used in order to re-set in order the data received from the memory so that it may correspond to an order that the command was received by the memory controller from the first. However, since other processors expect the data which is not as an order to some of processors expecting the data as an order, It should comment on what type of data order also being supported and getting by accepting necessity, and one [a switch] and turning off the switch of FIFO1404. The thing which needs this is from "it expects" about a processor receiving data in an order corresponding to the order which transmitted the command to the memory controller from the first.

[0077]

Since it may be received by the sequence to which the data from a memory does not correspond to the sequence of the origin to which a processor transmits a memory command with a memory controller, the 3rd sequence is stored in the data queue 1408. This sequence (this example 3, 2, 1) expresses an order that probably the data corresponding to the command sequences 1, 3, and 2 will be received in the data reading circuit 1406. A data queue sequence is computed by the re-ordering circuit 1400 based on the known waiting time related with the logic bank with various command queue sequences and memories. When a memory transmits data to a memory controller by the sequence (namely, 3, 1, 2) stored in the data queue 1408, Data is stored in the reading data buffer 1410, and based on the information and the data queue 1408 in FIFO1404, it is re-set in order so that it may transmit to a processor in an order corresponding to an order of the original command sequence, i.e., 1, 2, and 3.

[0078]

Drawing 15 is a block diagram of the re-ordering circuit 1500 in the memory controller constituted according to the specific embodiment of this invention. The re-ordering circuit 1500 contains the command queue 1502 which stores and re-sets in order the command received from the system processor. Using the command issue time constraint and the data bus use restrictions which were related with the command which tends toward the same logic bank in a memory, the command queue 1502 calculates the issue time of each command, publishes a command and removes the published command from cue.

[0079]

The data queue 1504 stores the data element showing the data generation time corresponding to the published memory command, and when the new data generation time to each new input to cue is calculated and a corresponding memory transaction is completed, it removes a cue entry.

[0080]

The comparator matrix 1506 performs a collision-detection function. The data generation time (transmitted via the multiplexer 1508) of the command which is [issue] ready from the command queue 1502 is compared with the data generation time of the command published before being expressed with the data queue 1504 by this function. Issue of a command will be postponed if a collision is detected.

[0081]

Drawing 16 is a still more detailed block diagram of the re-ordering circuit 1500 of drawing 15. The command queue 1502 stores 61-bit information about a memory command with the specific each including the six command queue elements 1602 as shown in the diagram of drawing 17. The command field 1702 contains 40 bit-memory command packet which specifies a memory command. The command issue time (Cd) field 1704 is the 6-bit field, and displays the delta time of the clock cycle before a command is published. The value of the field 1704 is determined by the above-mentioned limiting circuit, and is related to the newest memory command corresponding to the same logic bank in a memory. That is, the value of Cd field expresses the waiting time between two commands to the same bank. The information about waiting time required for each bank is stored in a limiting circuit, and most is determined by the physical characteristic of a memory. Although Cd field is determined to each clock cycle within a command queue, there are some exceptions. For example, the waiting time between the continuous commands to the same logic bank is not changed. Therefore, when Cd field for the command turned to the specific bank becomes zero and is not published, the decrement of the Cd field of other commands to the same bank is not carried out until the first command is published.

[0082]

The data-generation-time (Dd) field 1706 is the 6-bit field, and shows the delta time of issue of the memory command from a command queue, and the clock cycle between corresponding data transfers. The Dd field 1706 must not be changed within a command queue. Command ID fields 1708 are 5 bit fields, and identify the command in the command packet 1702 to a one to one correspondence. This information is used with the information and data queue with which FIFO corresponds, in order that which may not miss which packet and which data corresponds to which packet so that the resequencing injury effect of a command or data may be done so. The logic bank (B) field 1710 is the field of a triplet, and it is identified to which logic bank in a memory the command packet is turned. Finally, the burst sign (Db) field 1712 is the 1-bit field, and the data required or written in shows that one or two clock cycles are occupied.

[0083]

If it returns and states to drawing 16, operation of the command queue is controlled by the command queue controller 1604. As the controller 1604 does not miss in which command queue element 1602 it is usable, it controls insertion to the specific cue element of the command which enters via the empty identification-of-position part 1606. The controller 1604 makes it easy to insert the information on a command queue element in the data queue 1504, when a corresponding command is published again. According to the specific embodiment, a command is inserted in the command queue 1502 regardless of the availability of the idle time slot on a command bus or a data bus.

[0084]

The Cd count is zero, and when there is no collision on a data bus, via the multiplexer 1608, a command is published by the command bus from at least one of the command queue element 1602 throats, and it deals in it. That is, the idle time slot on a command bus and/or a data bus must be identified. When a command is neither a lead nor a light, only a command bus time slot is needed (therefore, when you do not need a data bus resource). When commands are a lead and a light, the slot of both a command bus and a data bus is needed.

[0085]

The zero comparator 1610 of the controller 1604 is used in order to determine whether to be the first determination, i.e., Cd=0. The subtractor 1612 is used in order to subtract "1" from Cd count to each command queue element 1602 for every clock cycle, when it is not Cd=0 to the above-mentioned exception, i.e., specific command which cannot be published. In that case, the cue controller 1604 emits the mask signal (M) which prevents Cd count to all the commands to the same logic bank carrying out a decrement using Cd field and B field to all the cue elements.

[0086]

When there are two cue elements which are Cd=0 according to the specific embodiment, the highest thing (for example, oldest thing) of a priority is published. Referring to drawing 18 below, the address shifter 1614 determines the priority of the command in cue so that it may discuss in detail. According to other specific embodiments, a new command reaches a command queue, and when the Cd count is already zero, it is directly transmitted to a memory via the multiplexer 1608. A new command is stored in the command queue element 1602, when the Cd count is not zero, or when there are other commands from which the priority was stored in the command queue in the higher rank Cd=0. However, when a command queue is empty, a new command is published immediately (when Cd is equal to zero).

[0087]

About a read command or a write command, a collision is detected using Dd field and Db field of the command queue element 1602 including the command which is [issue] ready. The data generation time and temporal duration corresponding to a command are transmitted to the comparator matrix 1506 via the multiplexer 1508, and the multiplexer 1508 is controlled by the cue controller 1604. That is, the cue controller 1604 controls the multiplexer 1508, in order for command issue time, i.e., Cd, to transmit the data generation time and temporal duration (1 or two clock cycles) of a cue element which are zero. Temporal duration is 1 or 2 clock cycles, and this is obtained when the adding machine 1616 draws "0" (one clock cycle is expressed) or "1" (two clock cycles are expressed) to Dd+1 by adding Db bit to the data generation time Dd. Next, data generation time and temporal duration are the comparator matrices 1506, and are compared with the data generation time of a command and temporal duration which were published before [five] which was stored in the data queue 1504. According to the specific embodiment, the comparator matrix 1506 includes 2x10 parallel comparator matrices.

[0088]

Drawing 18 is a block diagram of the specific embodiment of the address shifter 1614 of drawing 16. As mentioned above, the address shifter 1614 determines the priority of a command. As for this, a new command is inserted in the arbitrary free command queue elements 1602 as mentioned above according to the empty position recognition part 1606. The address of the command queue element 1602 with which a new command is inserted is inserted in the first empty position (A0 - A5) by the top priority. As a result, A0 position of an address shifter stores the oldest unissued cue element address for commands. When a command is published from a command queue, the entry to which the address shifter 1614 corresponds is removed, and the address for the commands of a low-ranking priority is changed into the position of the priority of a higher rank. As mentioned above, a command is published when Cd count to the command in a command queue reaches zero. However, when there are one or more commands of Cd=0, the oldest command, i.e., the command of the top [priority / which is shown by the position of the address in the address shifter 1614], is published.

[0089]

Including the five cue elements 1652, the data queue 1504 of drawing 16 includes the 12-bit information about the memory command published before as the each is illustrated by drawing 19. The data-generation-time (Dd) field 1902 is the 6-bit field which shows the delta time in the clock cycle between issue of the command from a command queue, and reception of corresponding data. The decrement of the Dd count to each data queue element 1652 is carried out for every clock cycle using one of the subtractors 1654 until the value reaches zero. Corresponding data is on a data bus at the time of Dd=0. Therefore, it will be understood that the data queue element 1652 which has Dd=0 at arbitrary time is only one. After Dd count reaches zero, the information within a corresponding data queue element is removed from the data queue 1504.

[0090]

Command ID field 1904 is the 5-bit field which identifies the issued command to which data corresponds to a one to one correspondence. This information is useful to re-set data in order so that a command may correspond in order of the origin first transmitted to the memory controller. Finally, the burst sign (Db) field 1906 is the 1-bit field which shows that data occupies one or two clock cycles.

[0091]

If it returns and discusses to drawing 16, the data generation time (Dd) and temporal duration over each of the data queue element 1652 as mentioned above, It is compared with Dd the command which is [issue] ready, i.e., for the commands in the command queue 1502 of Cd=0, and temporal duration by the comparator matrix 1506. Temporal duration is 1 or 2 clock cycles.

This is obtained when the adding machine 1616 draws "0" (one clock cycle is expressed) or "1" (two clock cycles are expressed) to Dd+1 by adding Db bit to the data generation time Dd.

If it turns out that there is no collision on a data bus by comparison, a command will be published from a command queue.

[0092]

The data queue controller 1658 controls operation of the data queue 1504. The empty identification-of-position part 1660 makes it easy to insert new data queue element information into the data queue element 1652 with the command queue controller 1604. It is made easy that an empty identification-of-position part removes information from the data queue element 1652 when corresponding memory access is completed again. The zero comparator 1662 and the burst identification part 1664, It is used for determining whether data transfer stops whether Dd to one of the data queue elements 1652 becomes zero, or when occupying [when] a data bus, and when, so, corresponding information should be removed from a data queue.

[0093]

According to another specific embodiment of this invention, collision detection becomes still more complicated through the comparator of two-dimensional array, and use of a multiplexer. this approach -- above-mentioned one-dimensional approach -- further -- silicon -- it is intensive and not only one element over the command which was [issue] ready but all the elements in a command queue are seen. This performs scheduling of a command about an order of the data packet on a data bus command [not only / that was published before / not only].

[0094]

It must be compared in order to investigate whether each combination of two continuous stages in the issue schedule portion of a command pipe can insert a new command among them in order to insert a new command. This comparison actually determines the range by which a command is inserted and in which it deals. This range is as being shown below.

[0095]

Length $T_{cstart}=t_{cA}+CLEN_{A}$ (1) of a $CLEN_{X}=$ command

$T_{cend}=t_{cB}$ (2)

[0096]

Here, t_{cA} and t_{cB} are the issue time of the continuous pipeline elements A and B. In advance of the pipeline element B therefore, the issue time of the pipeline element A is the lower one of an in [two]. If there is insertion, naturally between the elements of A and B, at least one open slot must be. It follows. : [0097]

$N=T_{cend}-T_{cstart}+1$ (3)

(Here, it is the number of the issue slots between the N= elements AB)

$LEN<=t_{cb}-t_{ca}-CLEN_{A}$ (4)

[0098]

In hardware, it is easy to mount the following conditions simply. : [0099]

$(t_{cB}-CLEN_{A})-(t_{cA}+CLEN_{A}) =>0$ (5)

[0100]

The starting point and the end point of the range also specify the possible range of the associated data slot. This range must be compared with each continuous element in a data pipe in order to investigate whether there is any overlap and what the new range becomes. Five different cases exist in this comparison.

[0101]

Case 0: In this case, the range drawn by data slot t_{dA} and t_{dB} is outside thoroughly [the range of the two continuous elements M and N]. In this case therefore : [0102]

$t_{dA}+CLEN_{A}=>t_{dN}$ (6)

Or it is $t_{dB}<=t_{dM}+DLEN_{M}$ (7) when it is the length of $DLEN_{X}=$ data.

[0103]

There is no possible data slot between the pairs of M and N.

【図１Ａ】

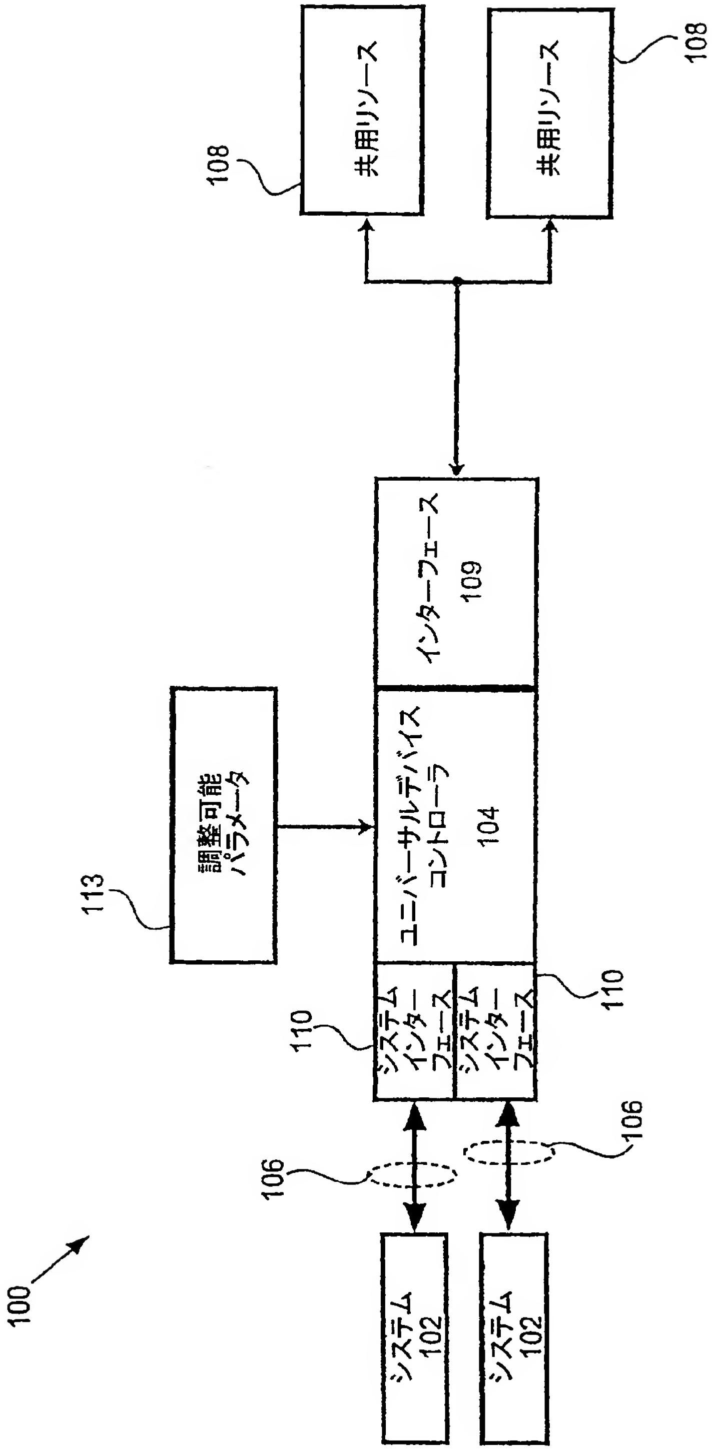

100
システム 102
システム 102
106
106
システム インターフェース
システム インターフェース
110
110
調整可能 パラメータ
113
ユニバーサルデバイス コントローラ 104
インターフェース 109
共用リソース
共用リソース
108
108

Fig. 1A

【図１Ｂ】

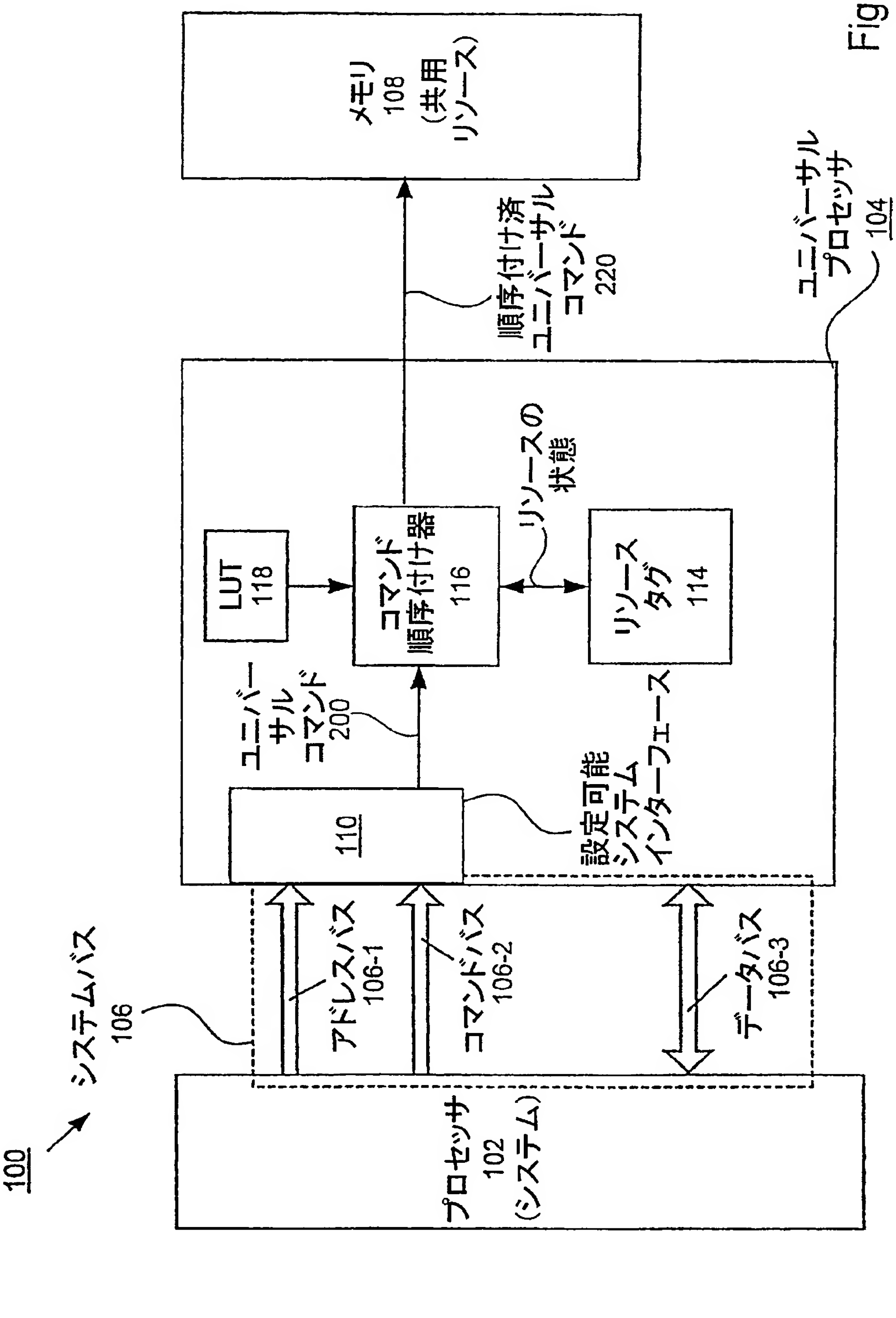

100
システムバス
106
プロセッサ
102
(システム)
アドレスバス
106-1
コマンドバス
106-2
データバス
106-3
110
設定可能
システム
インターフェース
ユニバー
サル
コマンド
200
LUT
118
コマンド
順序付け器
116
リソース
タグ
114
リソースの
状態
順序付け済
ユニバーサル
コマンド
220
メモリ
108
(共用
リソース)
ユニバーサル
プロセッサ
104

Fig. 1B

【図１Ｃ】

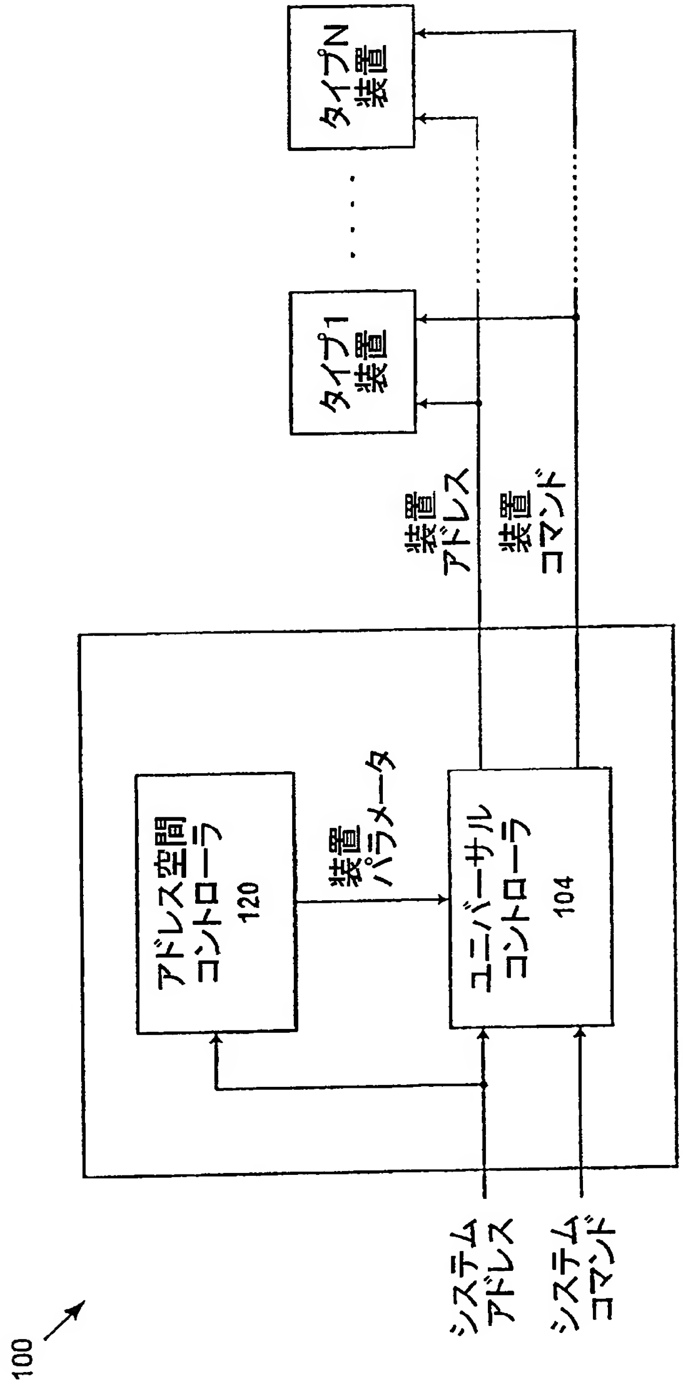

100
アドレス空間コントローラ
120
装置パラメータ
ユニバーサルコントローラ
104
システムアドレス
システムコマンド
装置アドレス
装置コマンド
タイプ1装置
タイプN装置

Fig. 1C

【図１D】

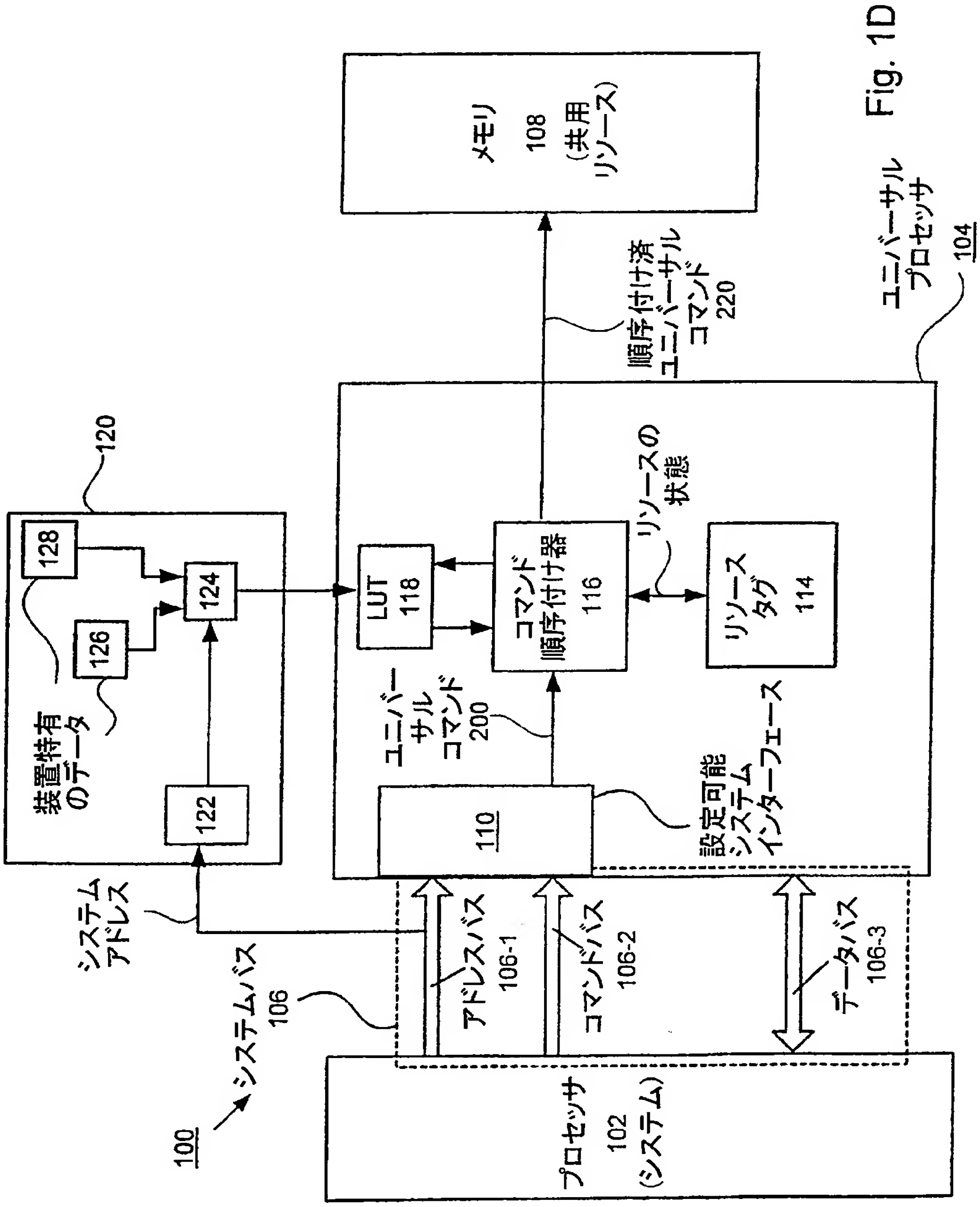

メモリ
108
(共用
リソース)
順序付け済
ユニバーサル
コマンド
220
ユニバーサル
プロセッサ
104
120
128
124
126
122
装置特有
のデータ
LUT
118
コマンド
順序付け器
116
リソースの
状態
リソース
タグ
114
ユニバー
サル
コマンド
200
110
設定可能
システム
インターフェース
システム
アドレス
システムバス
106
アドレスバス
106-1
コマンドバス
106-2
データバス
106-3
100
システム
プロセッサ
102
(システム)

Fig. 1D

【図１Ｅ】

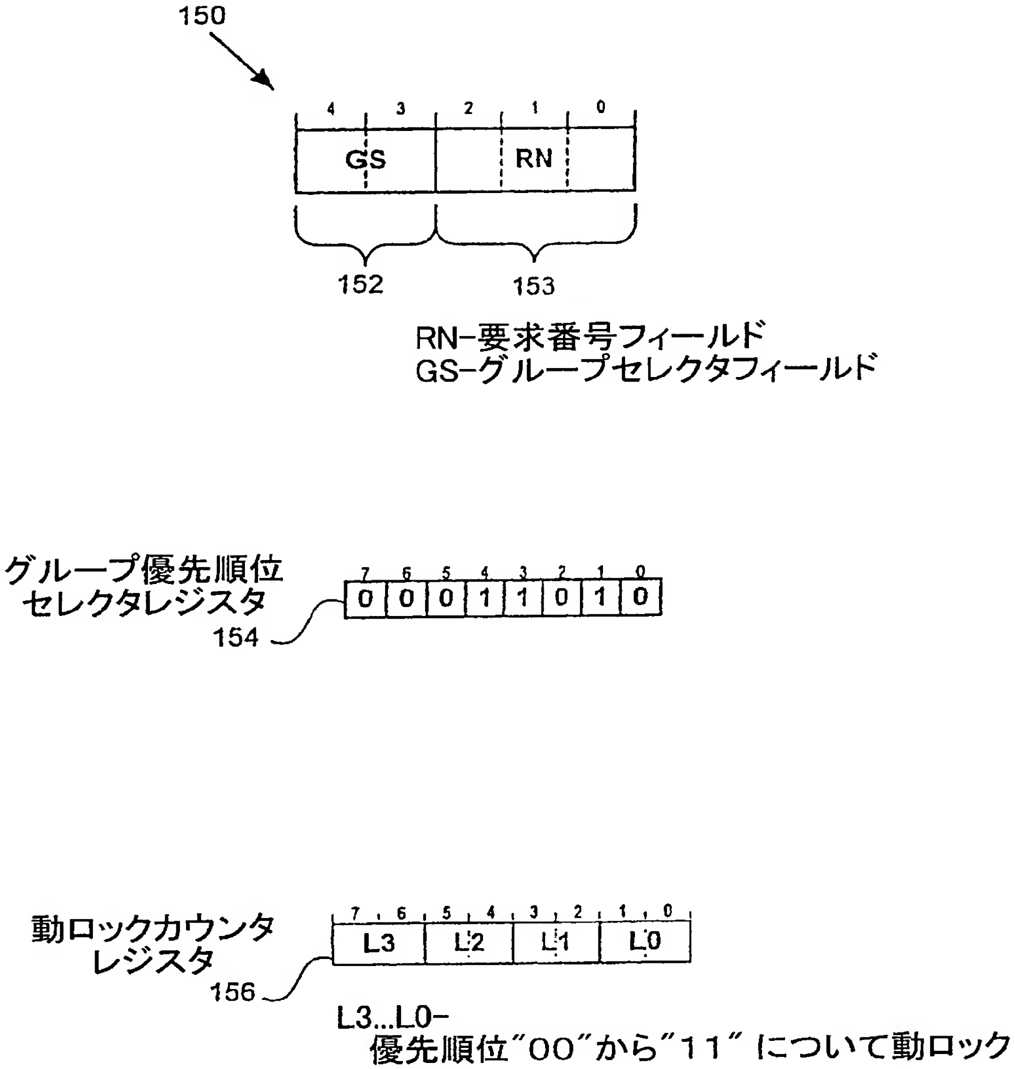

150
4 3 2 1 0
GS
RN
152
153
RN-要求番号フィールド
GS-グループセレクタフィールド
グループ優先順位
セレクタレジスタ
154
7 6 5 4 3 2 1 0
0 0 0 1 1 0 1 0
動ロックカウンタ
レジスタ
156
7 6 5 4 3 2 1 0
L3 L2 L1 L0
L3...L0-
優先順位"00"から"11" について動ロック

Fig. 1E

【図２Ａ】

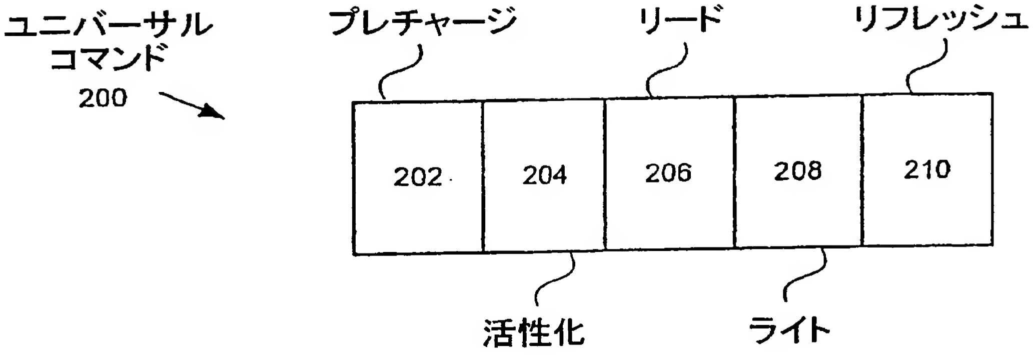


Fig. 2A

【図２Ｂ】

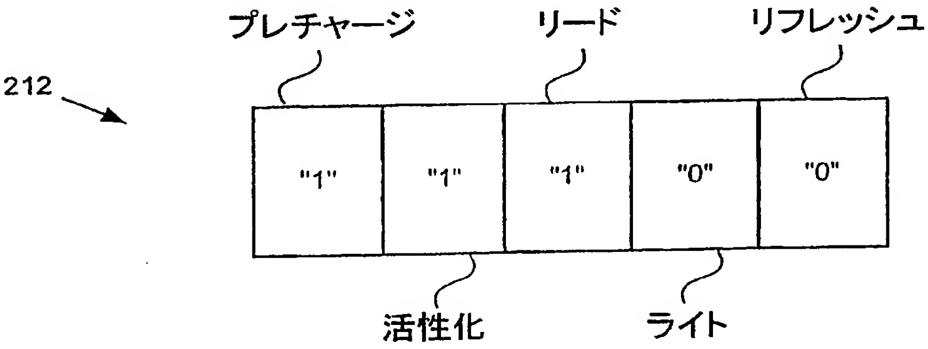


Fig. 2B

【図２Ｃ】

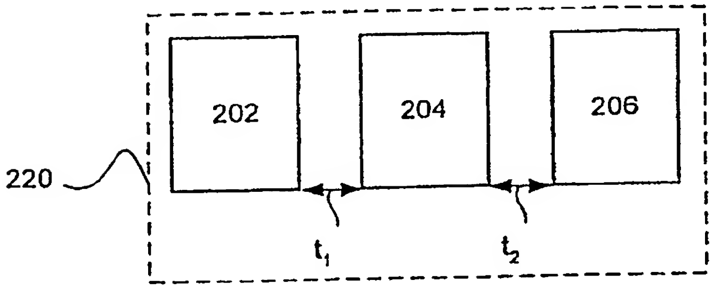


Fig. 2C

【図３】

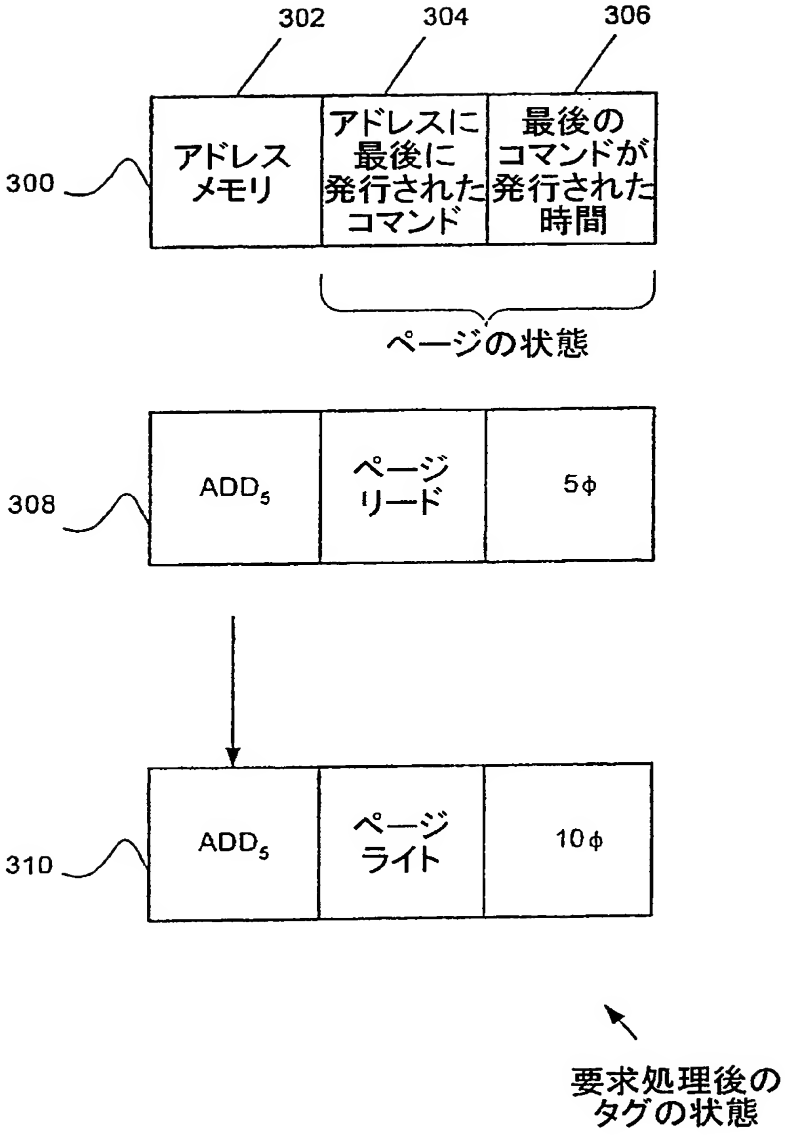

302
304
306
300
アドレスメモリ
アドレスに最後に発行されたコマンド
最後のコマンドが発行された時間
ページの状態
308
ADD$_5$
ページリード
5φ
310
ADD$_5$
ページライト
10φ
要求処理後のタグの状態

Fig. 3

【図４】

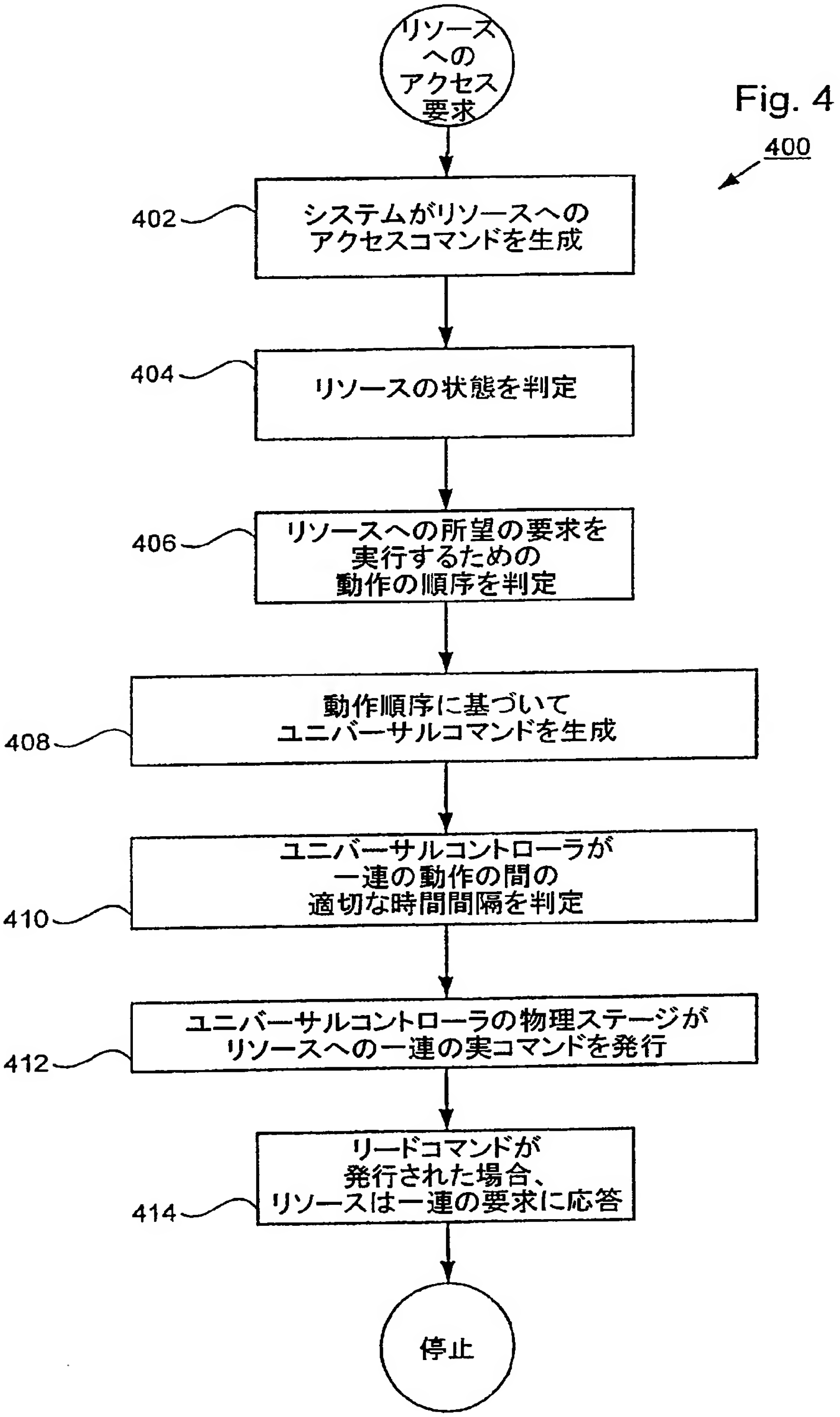


Fig. 4
400
リソースへのアクセス要求
402
システムがリソースへのアクセスコマンドを生成
404
リソースの状態を判定
406
リソースへの所望の要求を実行するための動作の順序を判定
408
動作順序に基づいてユニバーサルコマンドを生成
410
ユニバーサルコントローラが一連の動作の間の適切な時間間隔を判定
412
ユニバーサルコントローラの物理ステージがリソースへの一連の実コマンドを発行
414
リードコマンドが発行された場合、リソースは一連の要求に応答
停止

【図５】

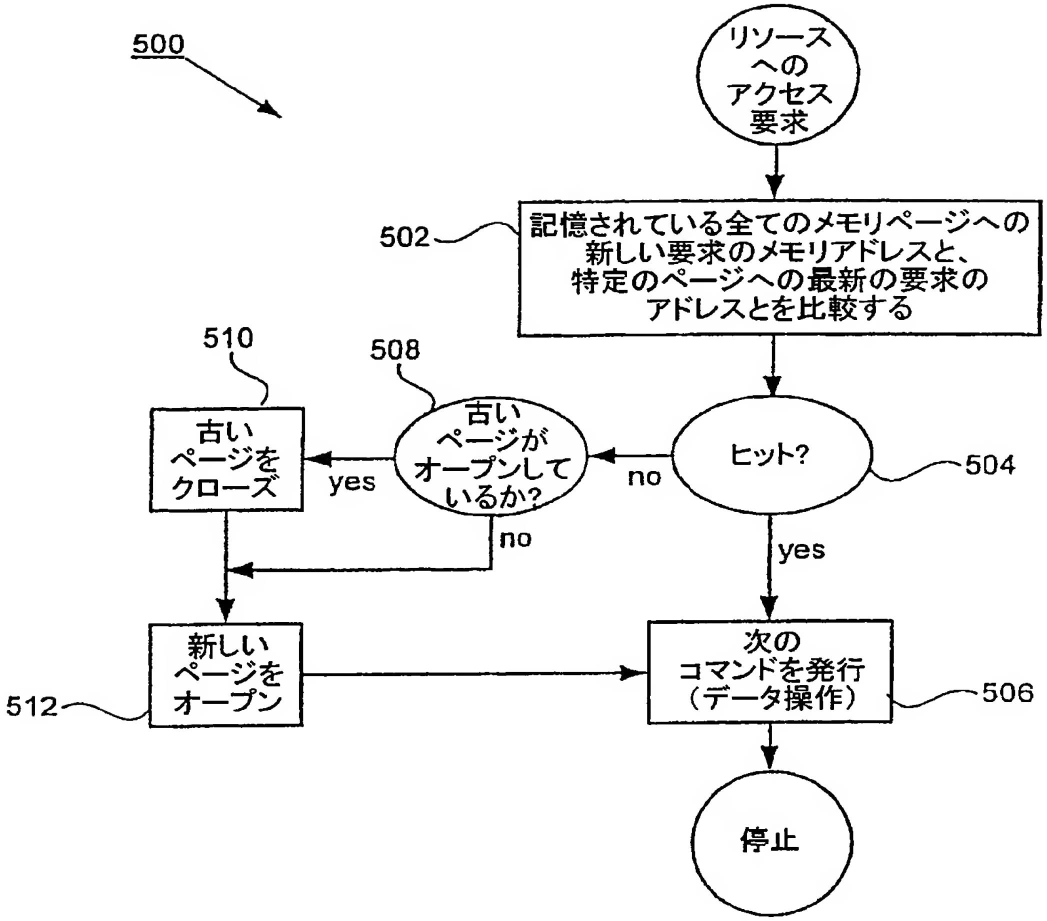

500
リソースへのアクセス要求
502
記憶されている全てのメモリページへの新しい要求のメモリアドレスと、特定のページへの最新の要求のアドレスとを比較する
504
ヒット?
no
508
古いページがオープンしているか?
yes
510
古いページをクローズ
no
512
新しいページをオープン
yes
506
次のコマンドを発行（データ操作）
停止

Fig. 5

【図６】

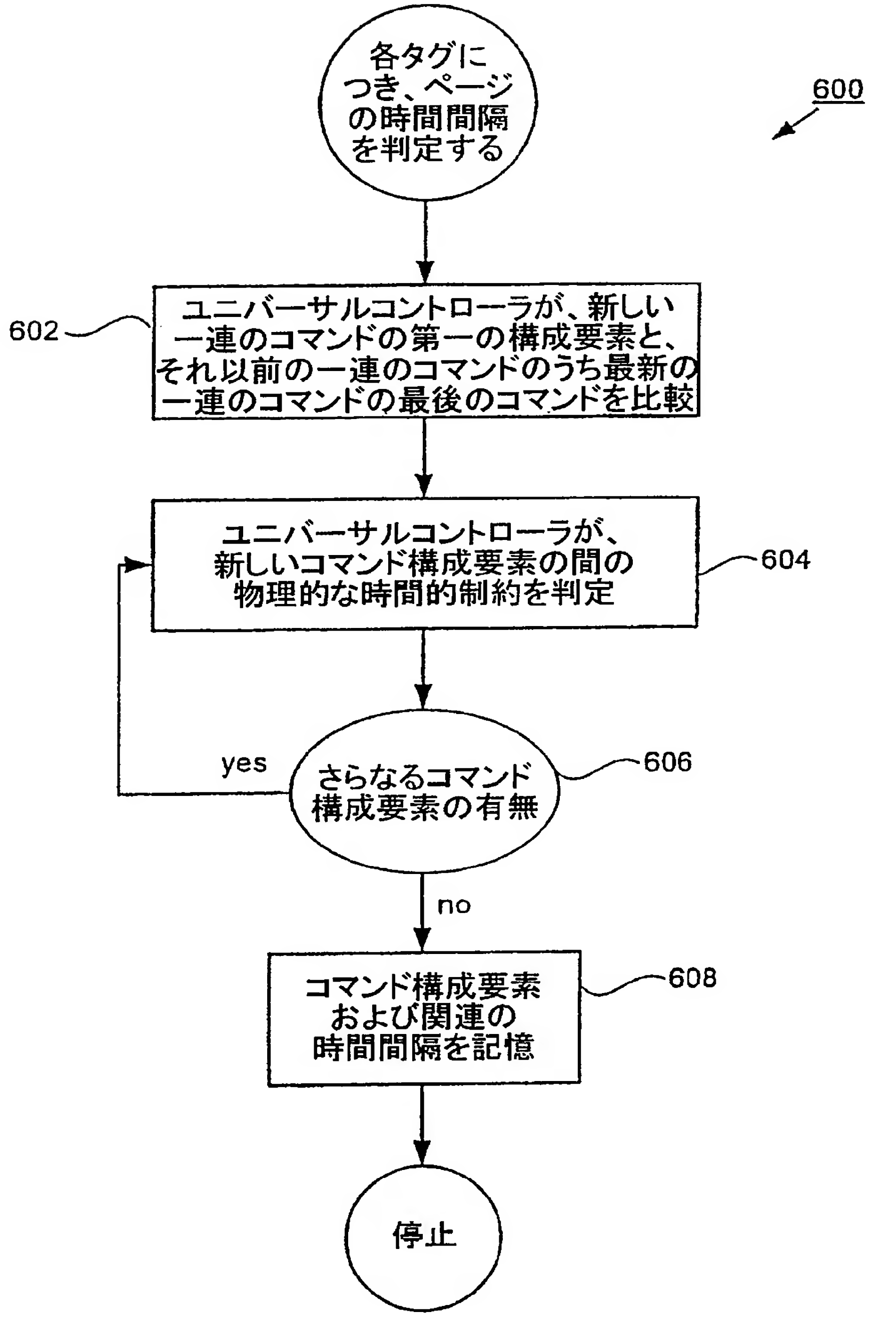

各タグにつき、ページの時間間隔を判定する
600
602
ユニバーサルコントローラが、新しい一連のコマンドの第一の構成要素と、それ以前の一連のコマンドのうち最新の一連のコマンドの最後のコマンドを比較
ユニバーサルコントローラが、新しいコマンド構成要素の間の物理的な時間的制約を判定
604
yes
さらなるコマンド構成要素の有無
606
no
コマンド構成要素および関連の時間間隔を記憶
608
停止

Fig. 6

【図７Ａ】

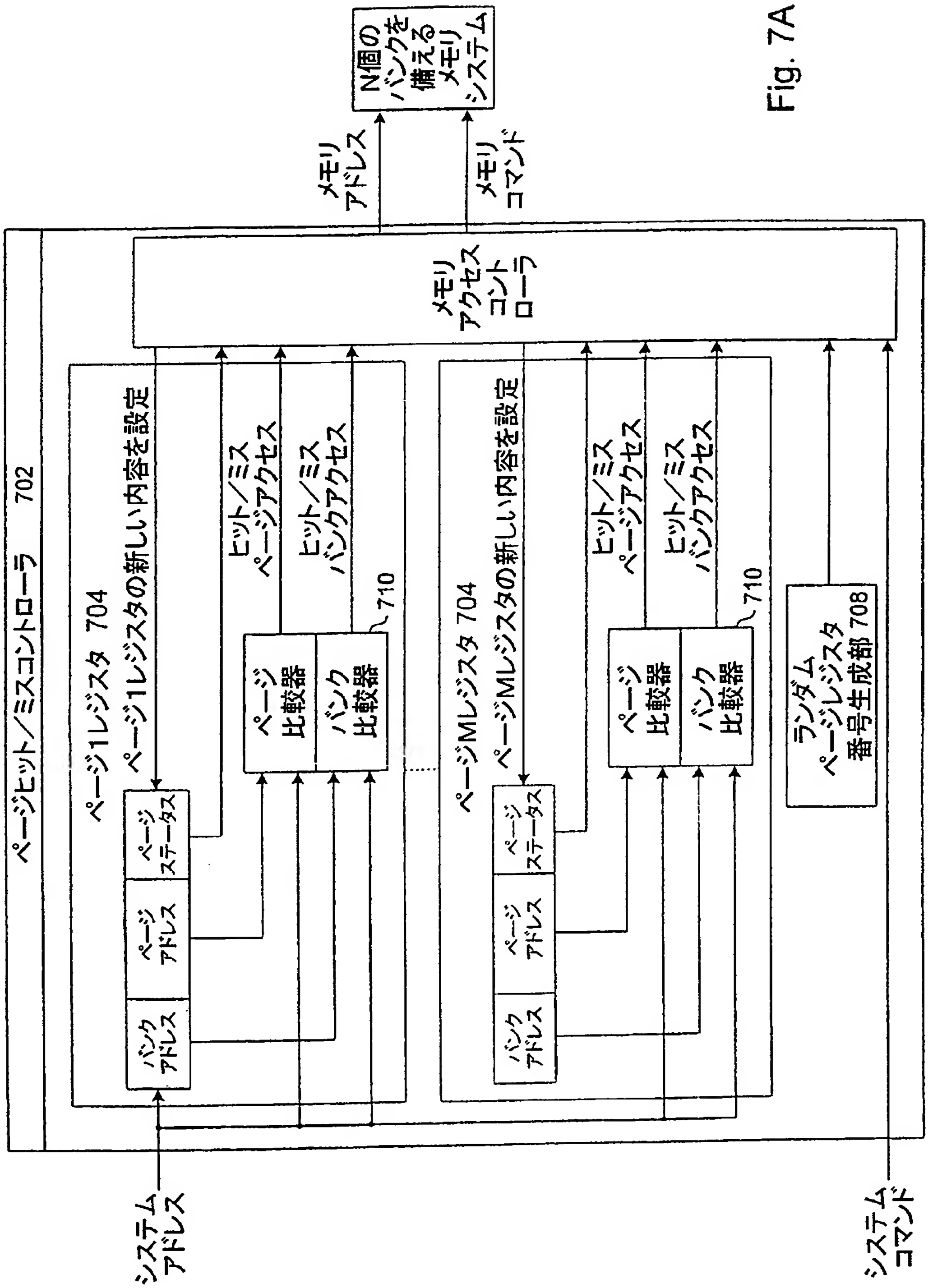

ページヒット／ミスコントローラ　702
ページ1レジスタ 704
ページ1レジスタの新しい内容を設定
ページ ステータス
ページ アドレス
バンク アドレス
ページ 比較器
バンク 比較器
710
ヒット／ミス ページアクセス
ヒット／ミス バンクアクセス
ページMレジスタ 704
ページMレジスタの新しい内容を設定
ページ ステータス
ページ アドレス
バンク アドレス
ページ 比較器
バンク 比較器
710
ヒット／ミス ページアクセス
ヒット／ミス バンクアクセス
ランダム ページレジスタ 番号生成部 708
メモリ アクセス コントローラ
メモリ アドレス
メモリ コマンド
N個の バンクを 備える メモリ システム
システム アドレス
システム コマンド

Fig. 7A

【図７Ｂ】

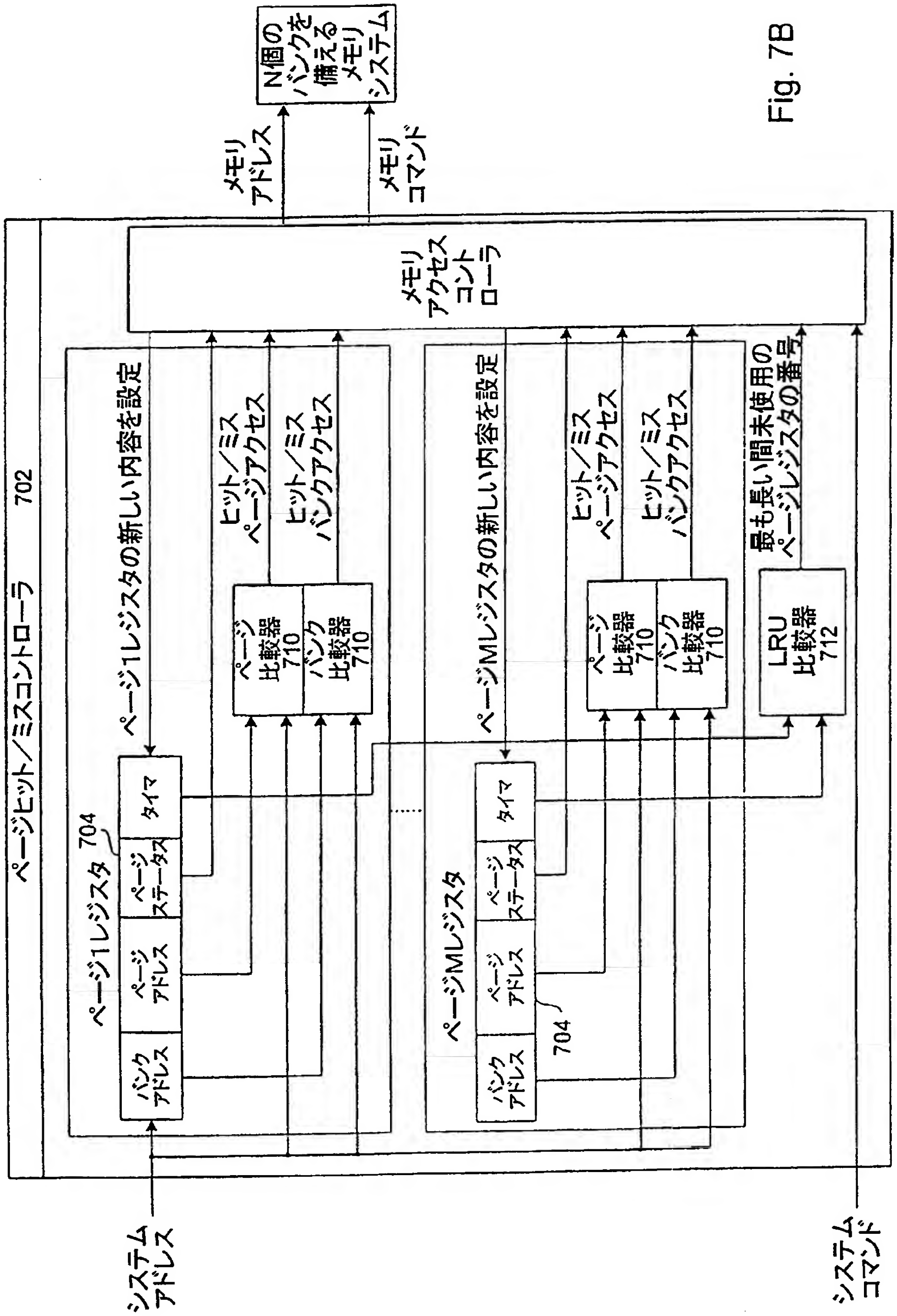

ページヒット／ミスコントローラ
702
ページ1レジスタ
704
バンク アドレス
ページ アドレス
ページ ステータス
タイマ
ページ1レジスタの新しい内容を設定
ヒット／ミス ページアクセス
ヒット／ミス バンクアクセス
ページ 比較器 710
バンク 比較器 710
ページMレジスタ
704
バンク アドレス
ページ アドレス
ページ ステータス
タイマ
ページMレジスタの新しい内容を設定
ヒット／ミス ページアクセス
ヒット／ミス バンクアクセス
ページ 比較器 710
バンク 比較器 710
最も長い間未使用の ページレジスタの番号
LRU 比較器 712
メモリ アクセス コント ローラ
メモリ アドレス
メモリ コマンド
N個の バンクを 備える メモリ システム
システム アドレス
システム コマンド

Fig. 7B

【図８】

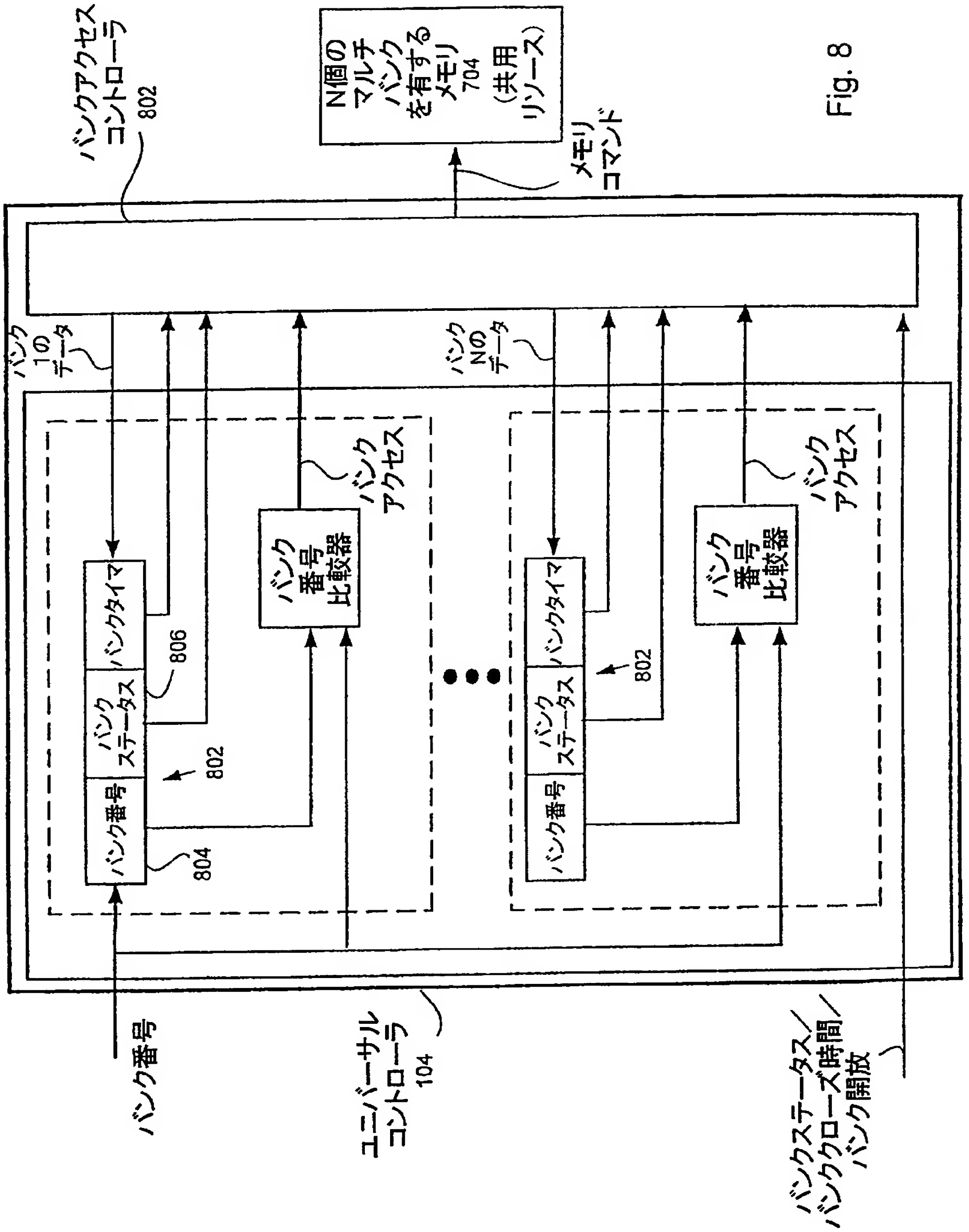
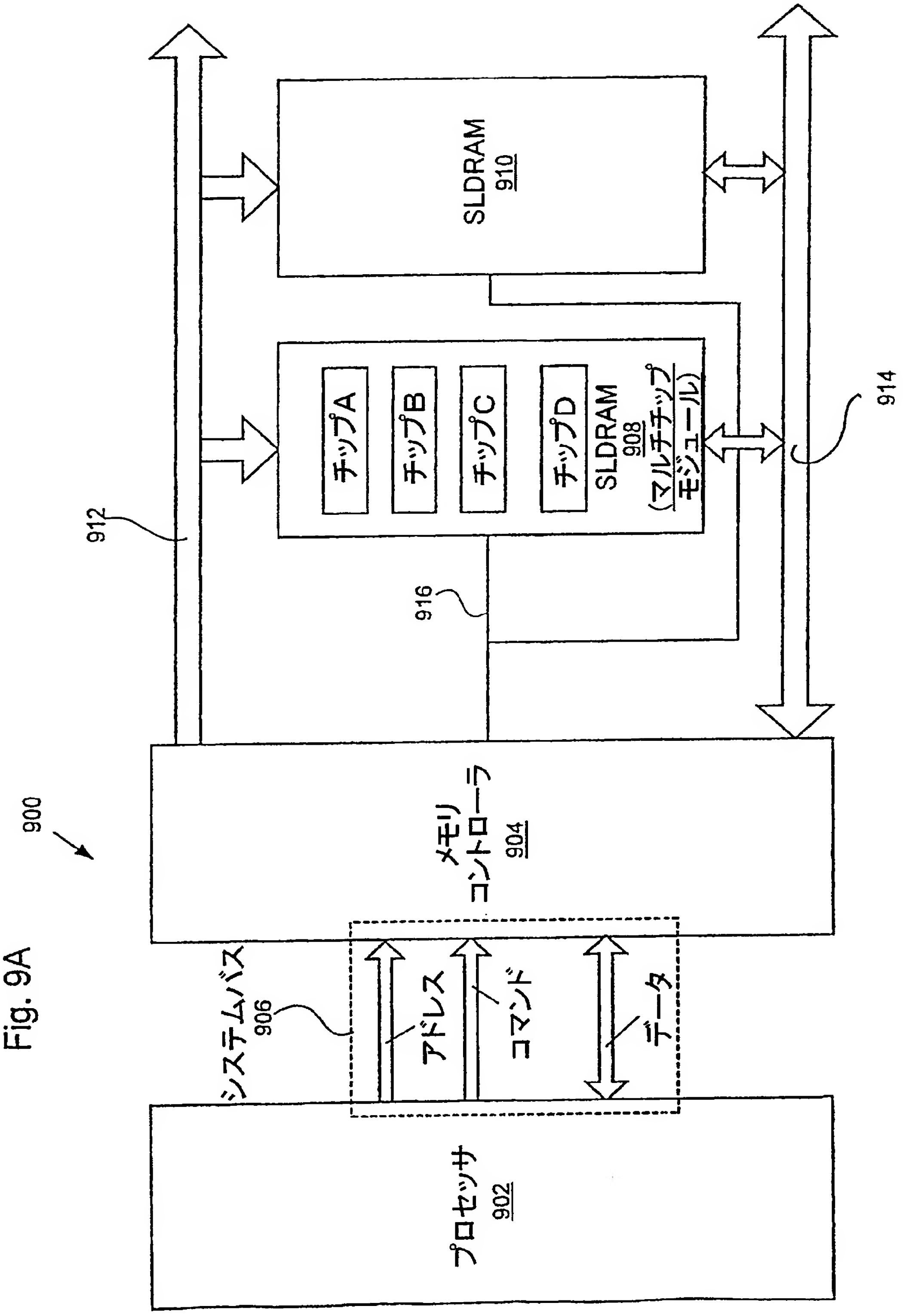

バンクアクセスコントローラ
802
N個のマルチバンクを有するメモリ
704
（共用リソース）
メモリコマンド
バンク1のデータ
バンクNのデータ
バンクタイマ
バンクステータス
バンク番号
806
802
804
バンク番号比較器
バンクアクセス
バンクタイマ
バンクステータス
バンク番号
802
バンク番号比較器
バンクアクセス
バンク番号
ユニバーサルコントローラ
104
バンクステータス／バンククローズ時間／バンク開放

Fig. 8

【図９Ａ】

Fig. 9A


900
プロセッサ
902
システムバス
906
アドレス
コマンド
データ
メモリ
コントローラ
904
912
916
チップA
チップB
チップC
チップD
SLDRAM
908
(マルチチップ
モジュール)
SLDRAM
910
914

【図９Ｂ】

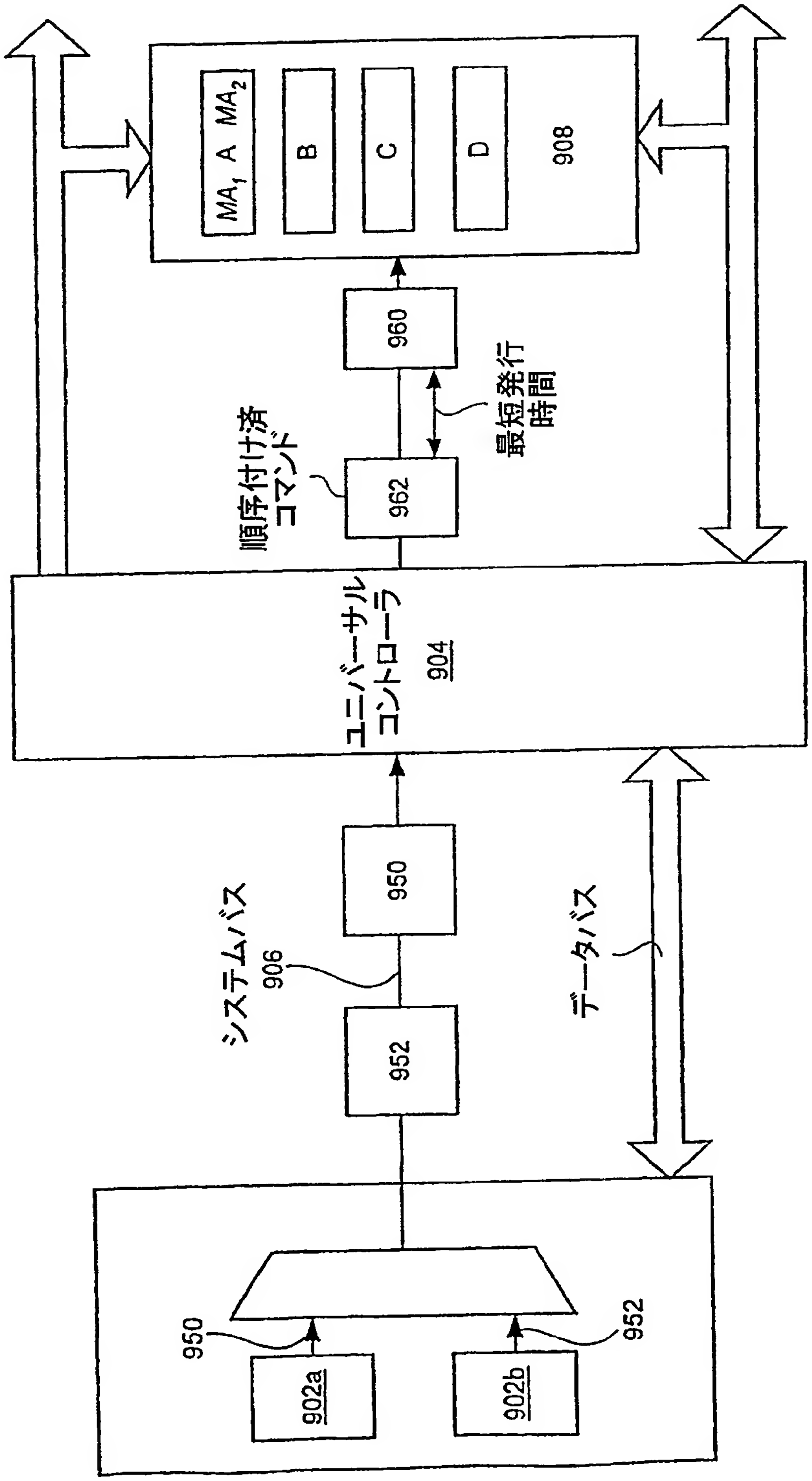

MA_1 A MA_2
B
C
D
908
960
順序付け済コマンド
962
最短発行時間
ユニバーサルコントローラ
904
950
システムバス
906
952
データバス
950
902a
902b
952

Fig. 9B

【図１０】

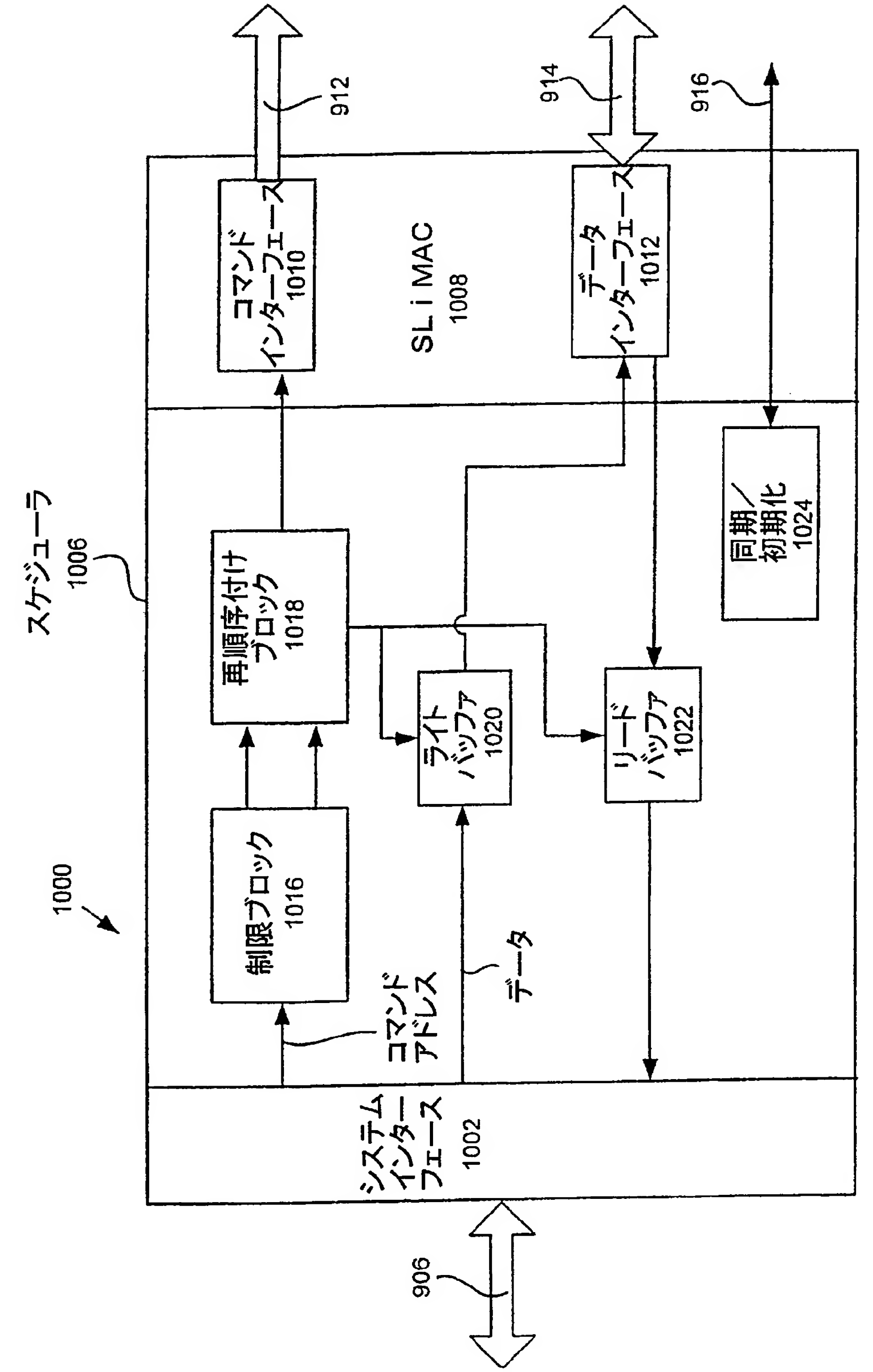

912
914
916
コマンド
インターフェース
1010
SL i MAC
1008
データ
インターフェース
1012
スケジューラ
1006
再順序付け
ブロック
1018
同期／
初期化
1024
ライト
バッファ
1020
リード
バッファ
1022
1000
制限ブロック
1016
コマンド
アドレス
データ
システム
インター
フェース
1002
906

Fig. 10

【図１１】

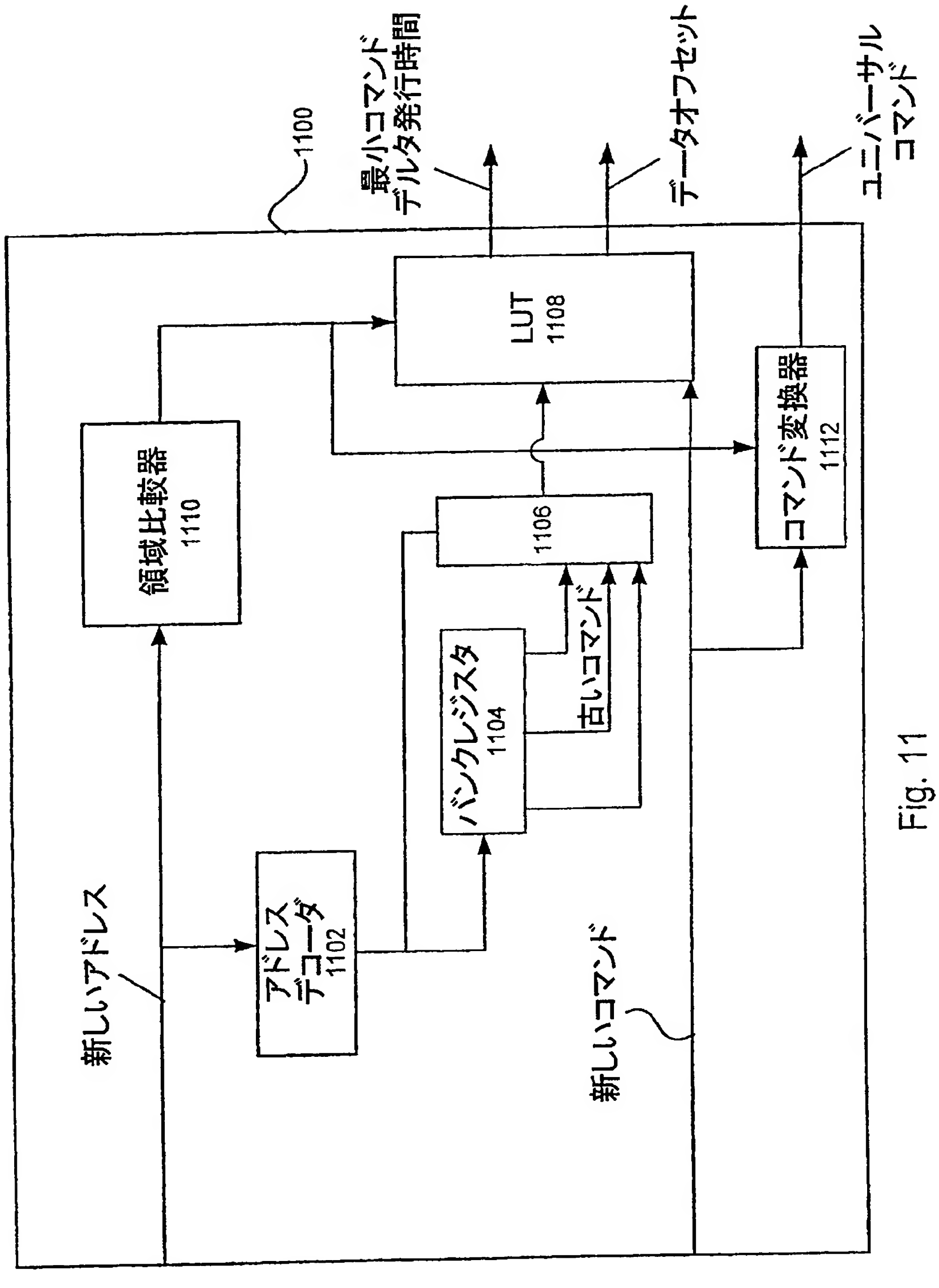

1100
最小コマンド
デルタ発行時間
データオフセット
ユニバーサル
コマンド
LUT
1108
領域比較器
1110
コマンド変換器
1112
1106
古いコマンド
バンクレジスタ
1104
アドレス
デコーダ
1102
新しいアドレス
新しいコマンド

Fig. 11

【図１２】

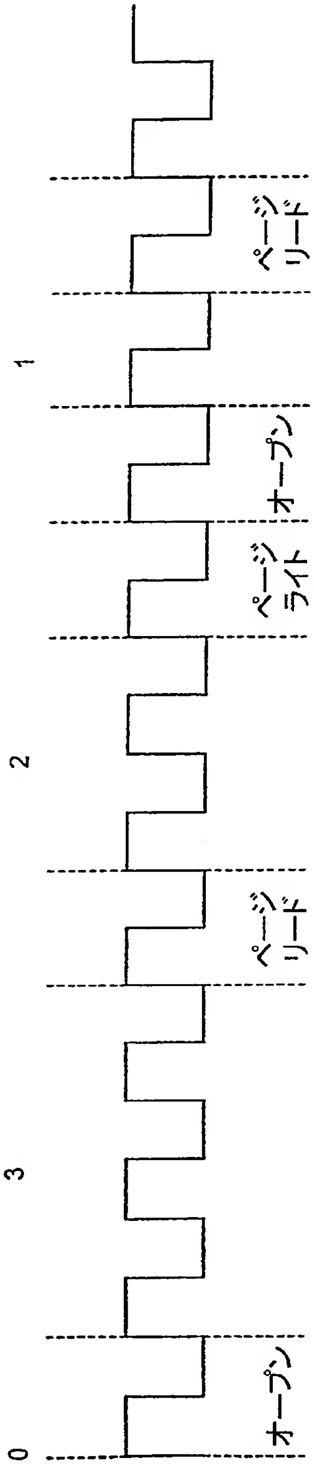


Fig. 12

【図１３Ａ】

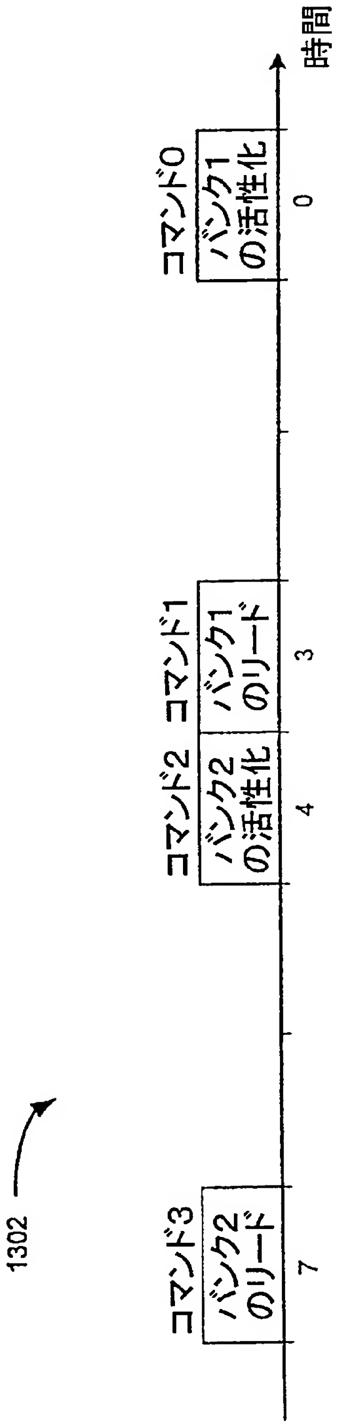

時間
コマンド0
バンク1
の活性化
0
コマンド1
バンク1
のリード
3
コマンド2
バンク2
の活性化
4
コマンド3
バンク2
のリード
7
1302

Fig. 13A

【図１３B】

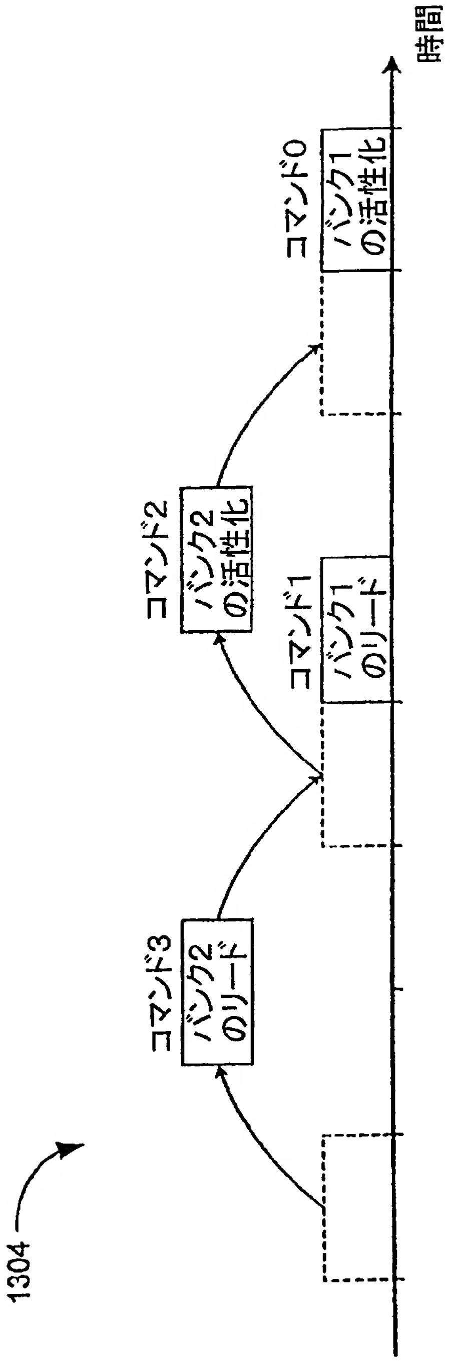

時間
コマンド0
バンク1
の活性化
コマンド1
バンク1
のリード
コマンド2
バンク2
の活性化
コマンド3
バンク2
のリード
1304

Fig. 13B

【図１３Ｃ】

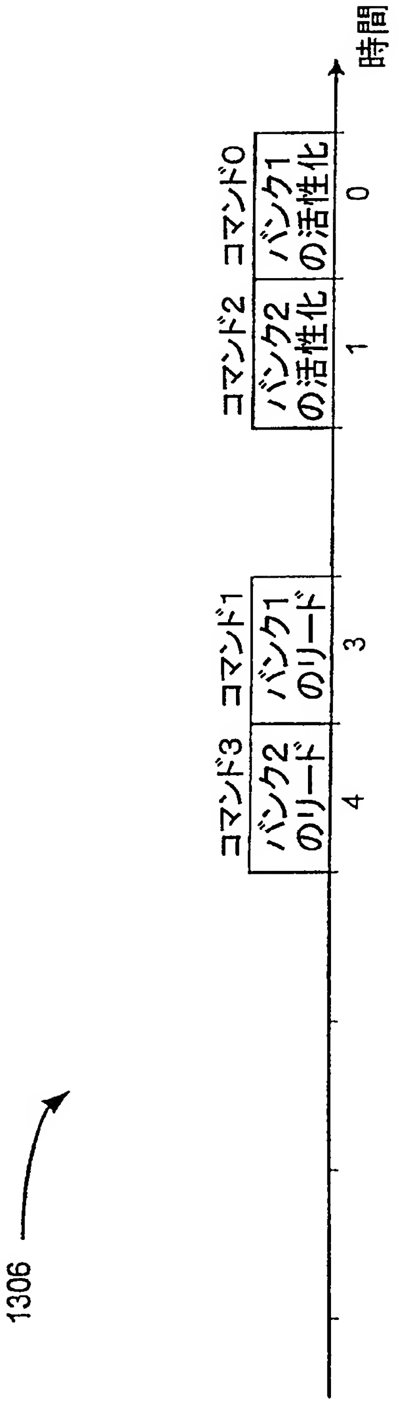

時間
コマンド0
バンク1
の活性化
0
コマンド2
バンク2
の活性化
1
コマンド1
バンク1
のリード
3
コマンド3
バンク2
のリード
4
1306

Fig. 13C

【図１４】

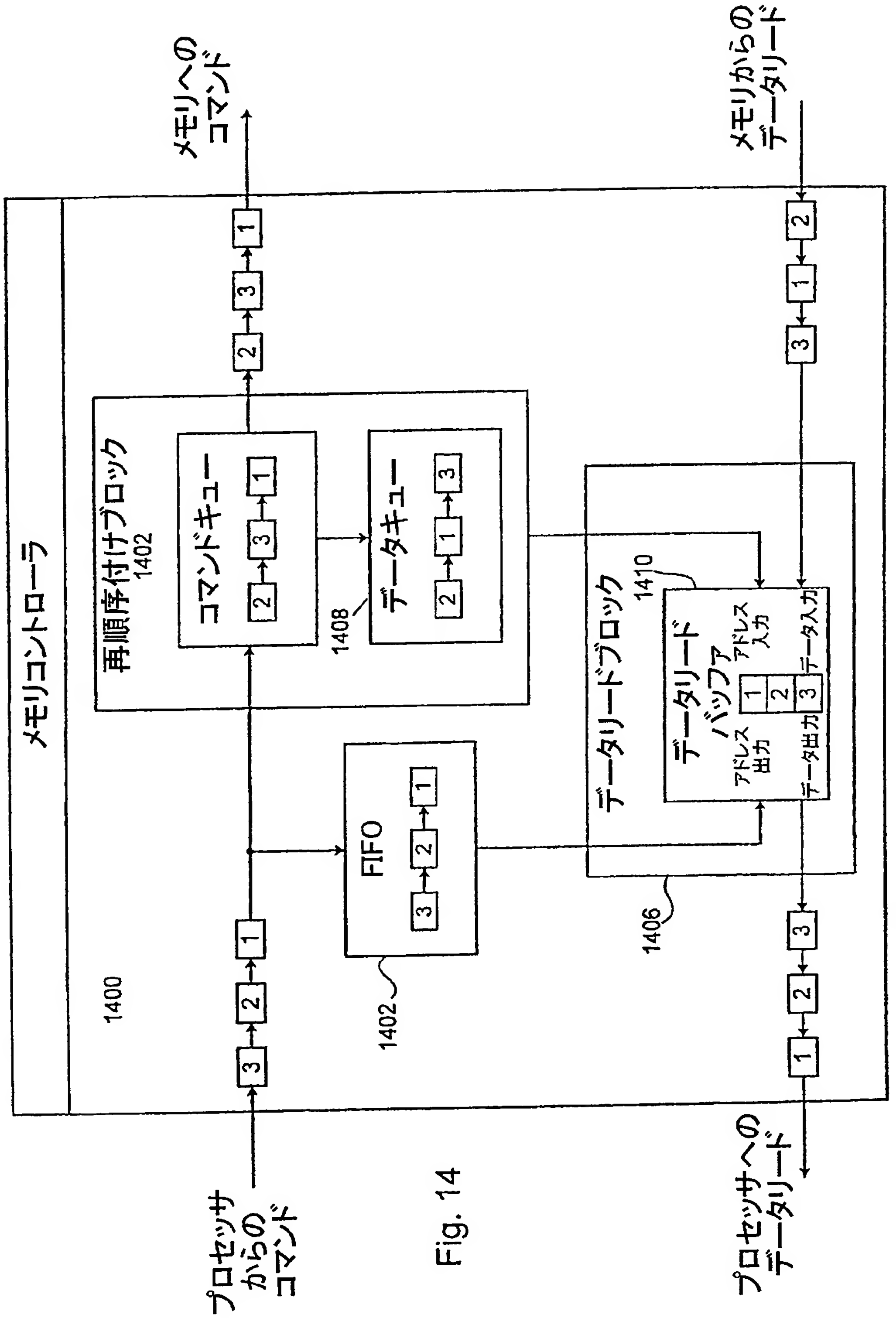

メモリコントローラ
1400
再順序付けブロック
1402
コマンドキュー
データキュー
1408
FIFO
1402
データリードブロック
1406
データリードバッファ
1410
アドレス入力
アドレス出力
データ入力
データ出力
メモリへのコマンド
メモリからのデータリード
プロセッサからのコマンド
プロセッサへのデータリード
1
2
3

Fig. 14

【図１５】

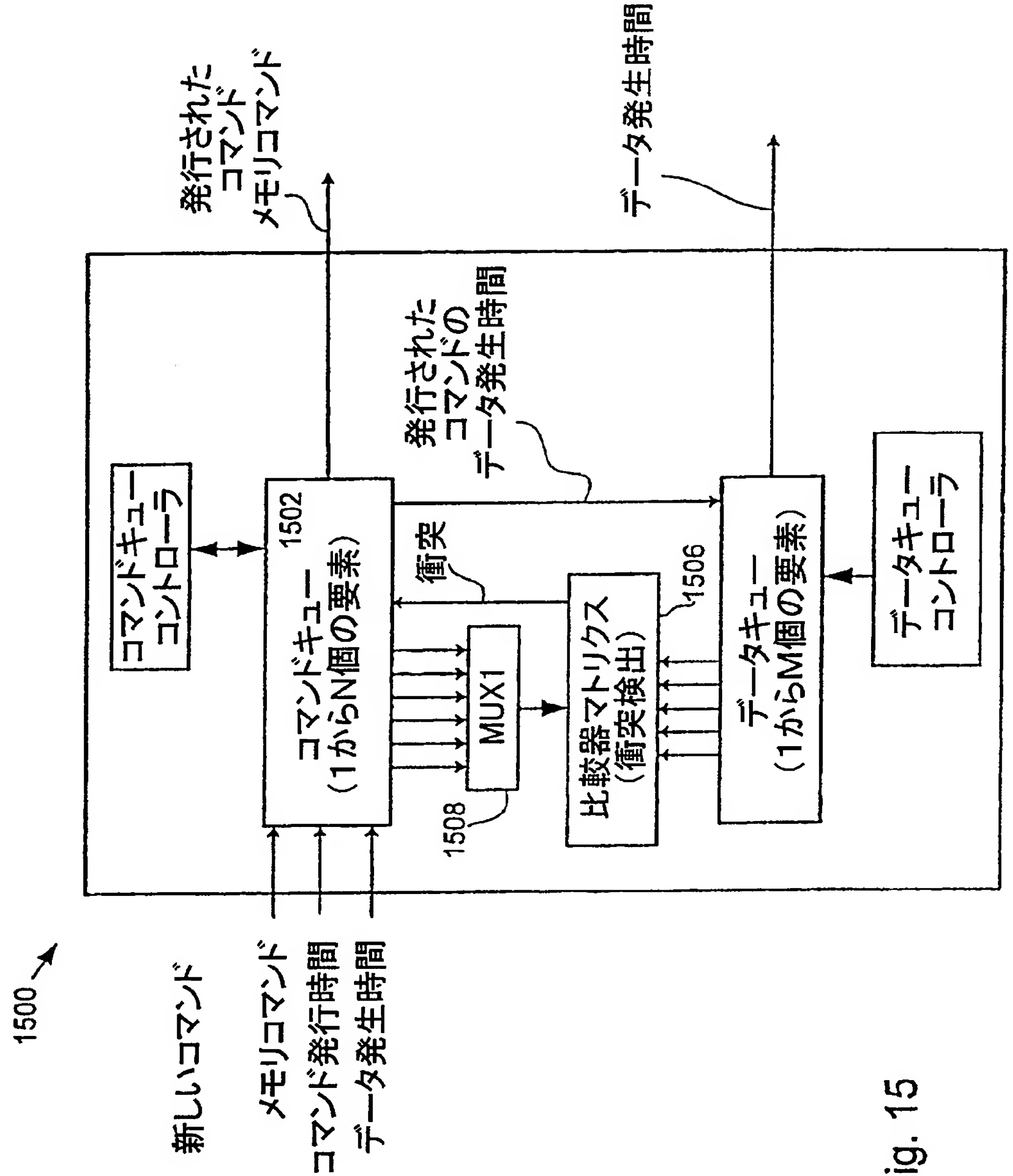

1500
新しいコマンド
メモリコマンド
コマンド発行時間
データ発生時間
発行された
コマンド
メモリコマンド
データ発生時間
コマンドキュー
コントローラ
1502
コマンドキュー
（１からN個の要素）
発行された
コマンドの
データ発生時間
衝突
MUX1
1508
比較器マトリクス
（衝突検出）
1506
データキュー
（１からM個の要素）
データキュー
コントローラ

Fig. 15

【図１６】

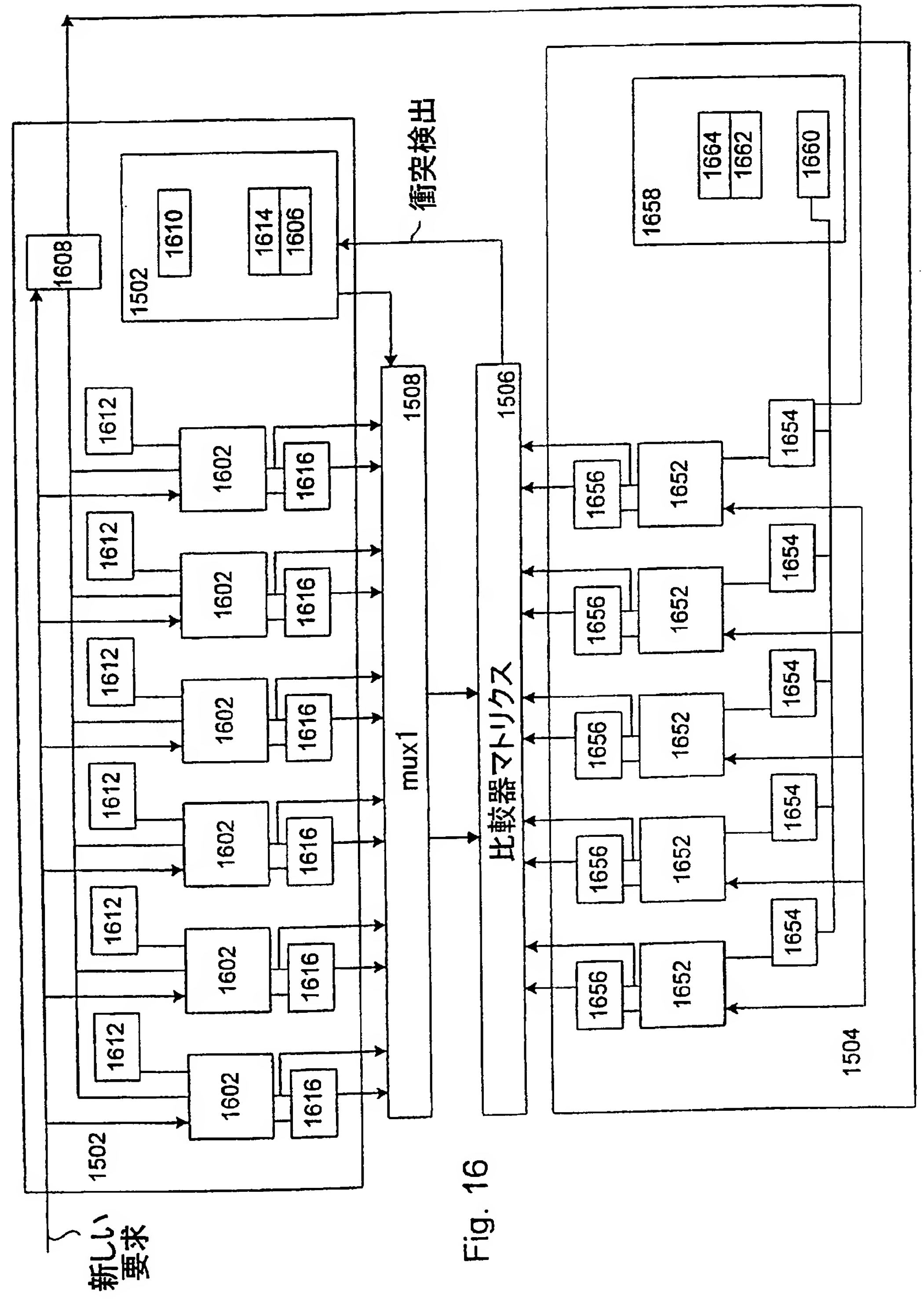

衝突検出
1502
1610
1614
1606
1608
1612
1602
1616
1508
mux1
1506
比較器マトリクス
1504
1658
1664
1662
1660
1654
1652
1656
新しい要求

Fig. 16

【図１７】

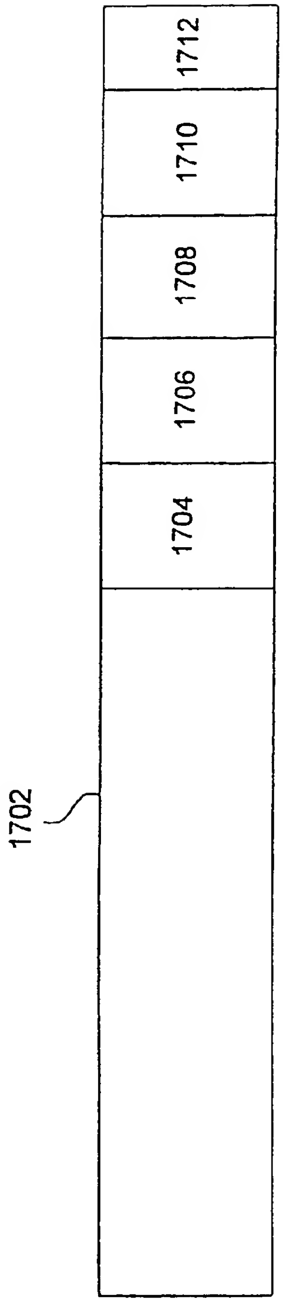


Fig. 17

【図１８】

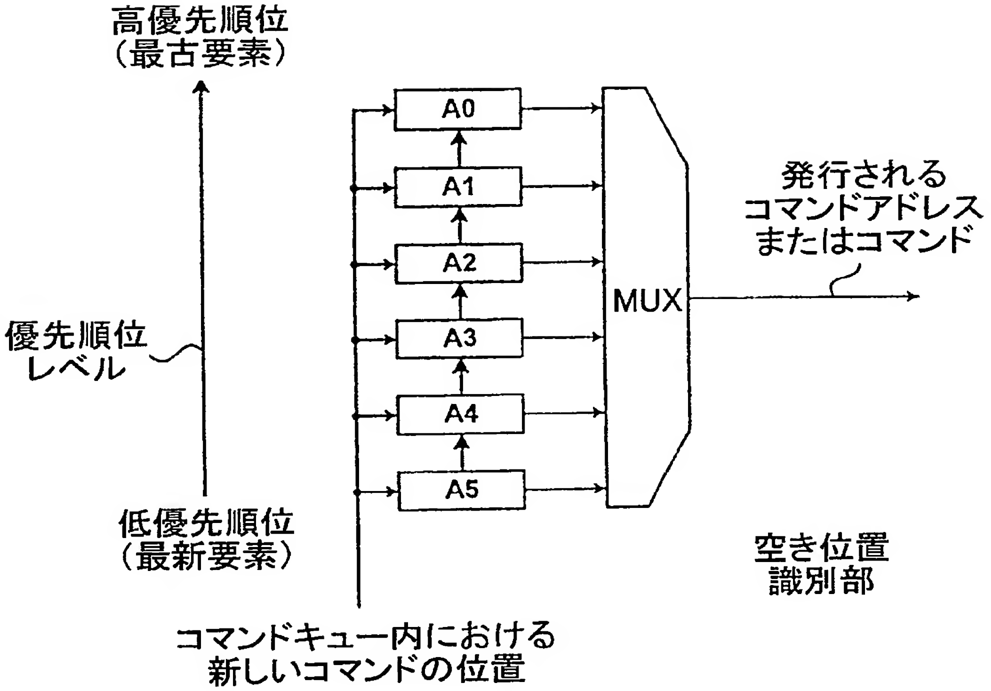

高優先順位
（最古要素）
優先順位
レベル
低優先順位
（最新要素）
A0
A1
A2
A3
A4
A5
MUX
発行される
コマンドアドレス
またはコマンド
空き位置
識別部
コマンドキュー内における
新しいコマンドの位置

Fig. 18

【図１９】

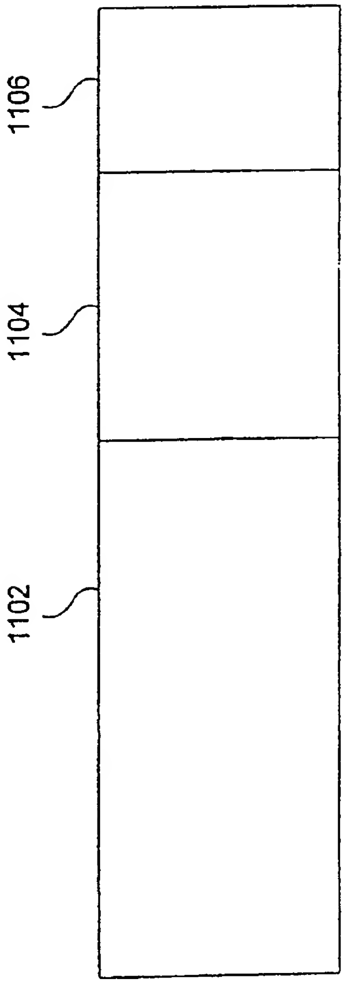


Fig. 19

【図２０】

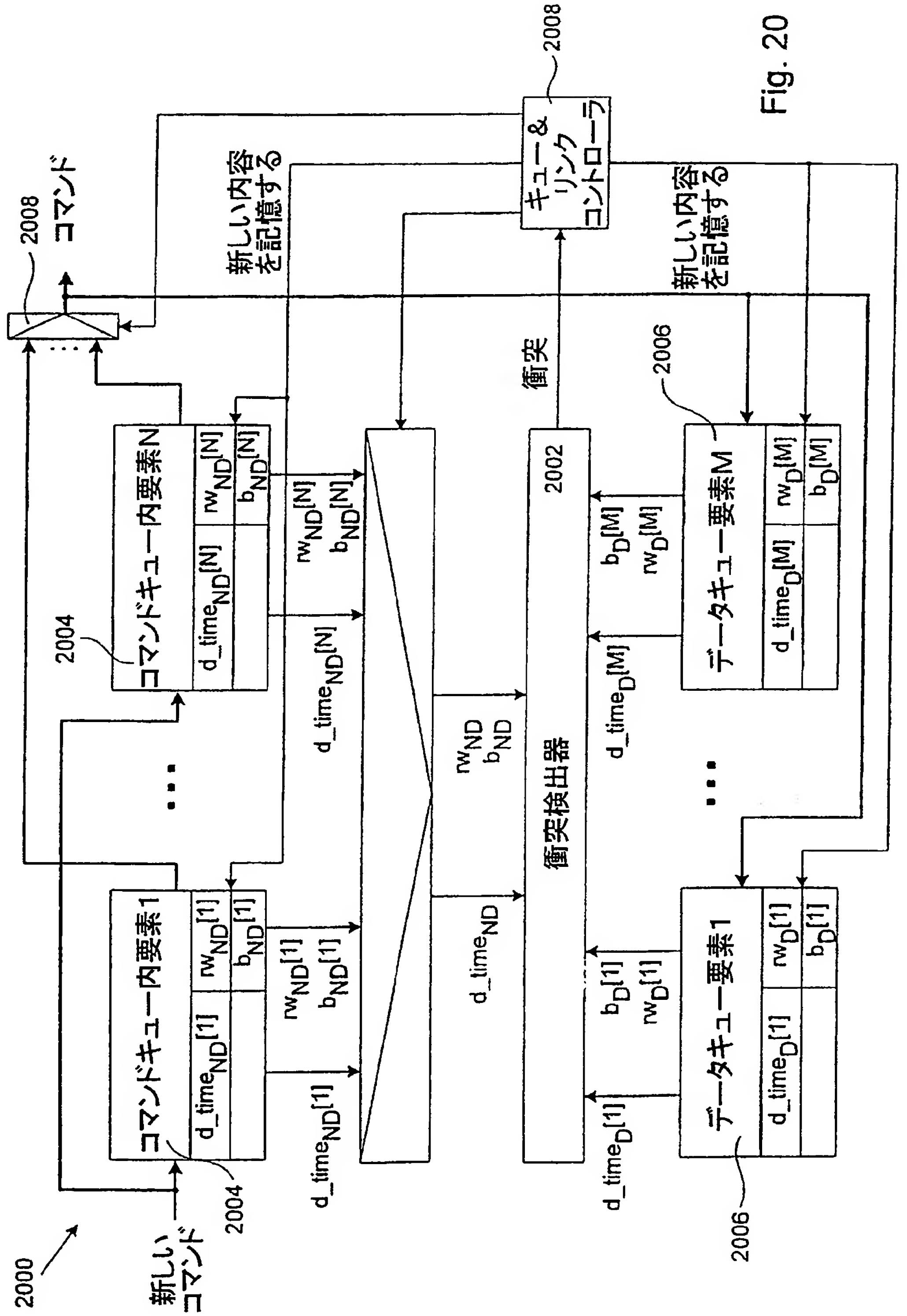

2000
新しいコマンド
2004
コマンドキュー内要素1
$d\_time_{ND}[1]$
$rw_{ND}[1]$
$b_{ND}[1]$
2004
コマンドキュー内要素N
$d\_time_{ND}[N]$
$rw_{ND}[N]$
$b_{ND}[N]$
2008
コマンド
新しい内容を記憶する
d_time_{ND}
rw_{ND}
b_{ND}
衝突検出器
2002
衝突
キュー＆リンクコントローラ
2008
新しい内容を記憶する
2006
データキュー要素M
$d\_time_D[M]$
$rw_D[M]$
$b_D[M]$
2006
データキュー要素1
$d\_time_D[1]$
$rw_D[1]$
$b_D[1]$

Fig. 20

【図２１】

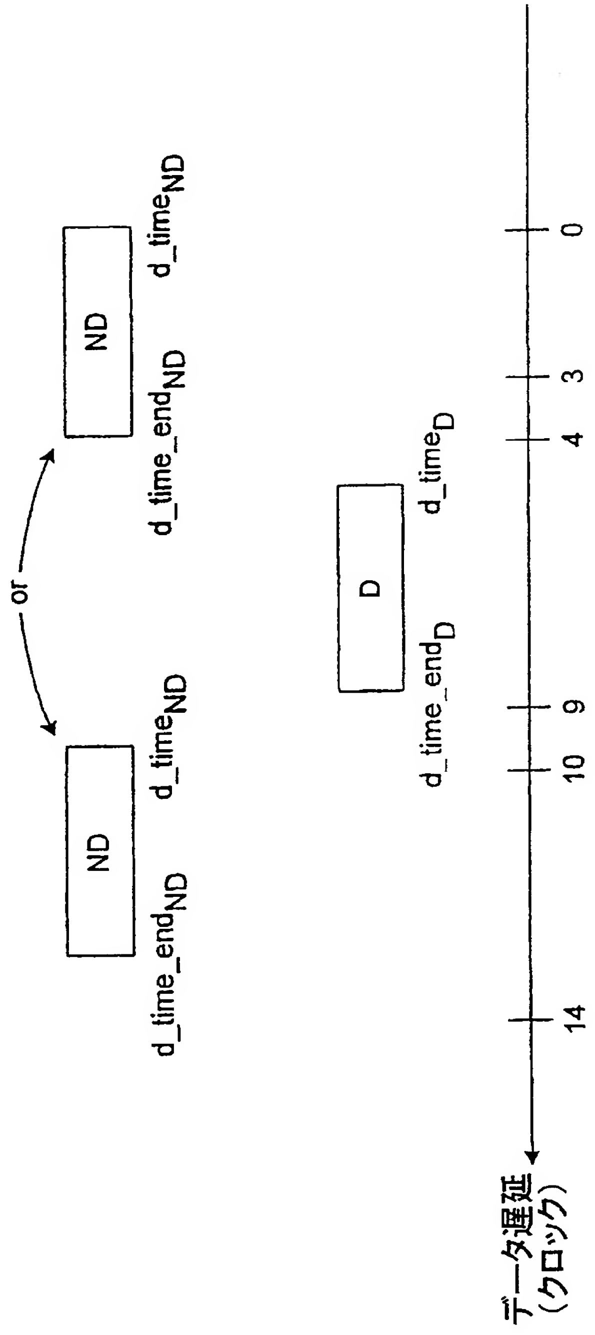

or
ND
d_time_end$_{ND}$
d_time$_{ND}$
ND
d_time_end$_{ND}$
d_time$_{ND}$
D
d_time_end$_D$
d_time$_D$
0
3
4
9
10
14
データ遅延
（クロック）

Fig. 21

【図２２】

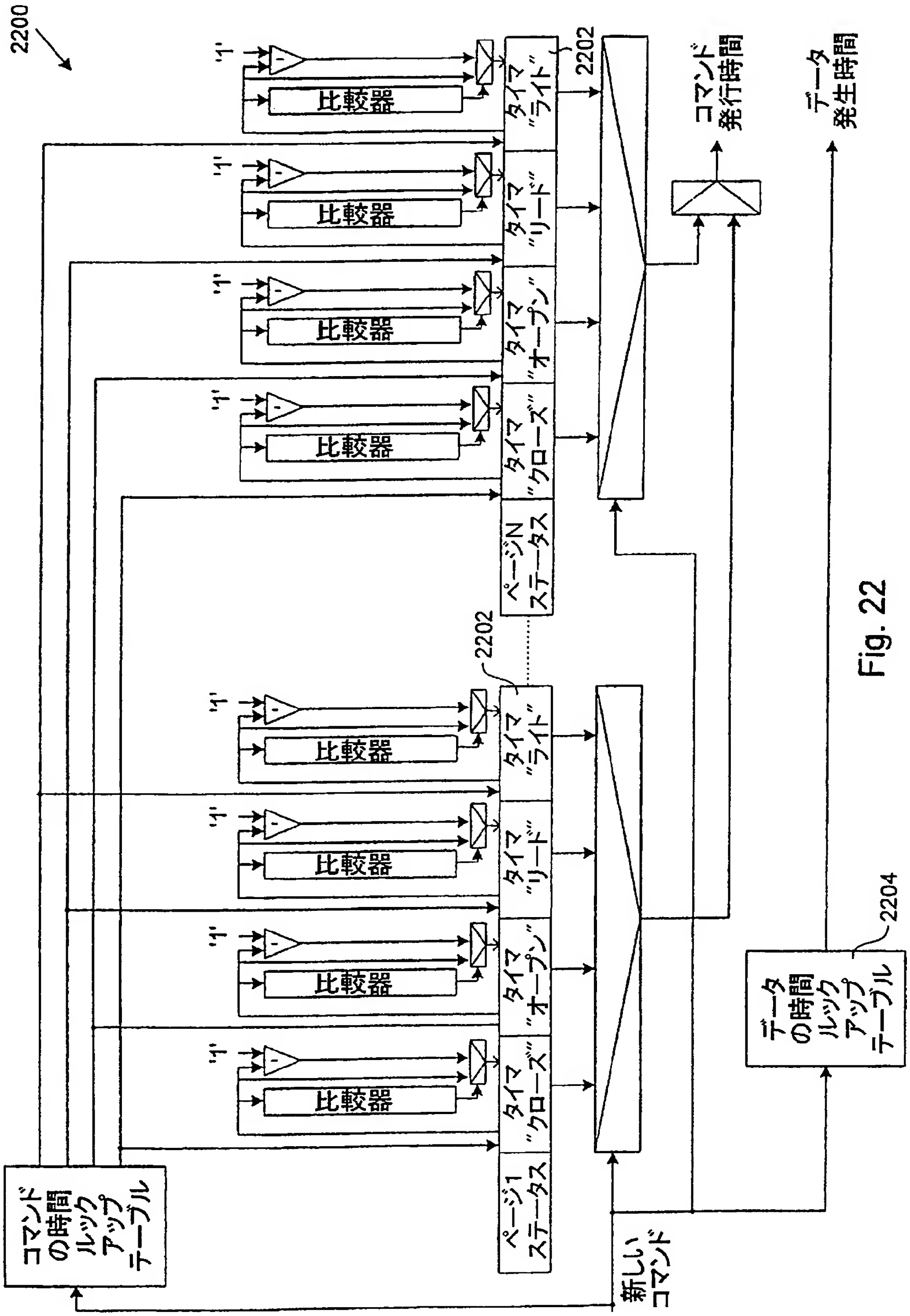

2200
'1'
比較器
タイマ"ライト"
タイマ"リード"
タイマ"オープン"
タイマ"クローズ"
ページNステータス
ページ1ステータス
2202
2202
コマンド発行時間
データ発生時間
コマンドの時間ルックアップテーブル
データの時間ルックアップテーブル
2204
新しいコマンド

Fig. 22

【図２３】

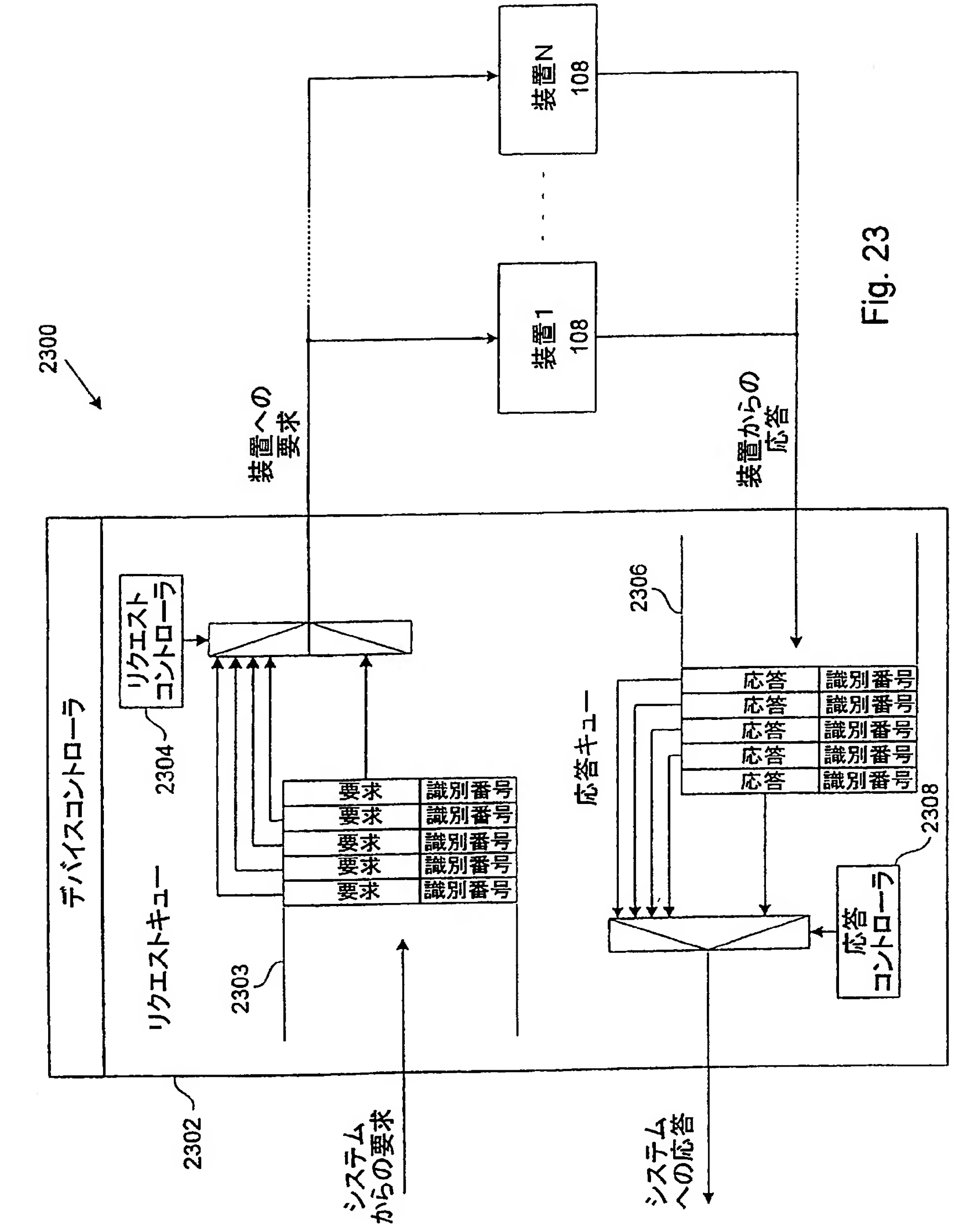

2300
デバイスコントローラ
リクエストキュー
リクエストコントローラ
2304
2303
要求
識別番号
装置への要求
装置1
108
装置N
108
装置からの応答
2306
応答キュー
応答
識別番号
応答コントローラ
2308
2302
システムからの要求
システムへの応答

Fig. 23

【図２４】

表 4

入力コマンド	領域		仮想バンク 0	1	2	3	代替	LUT 入力	ΔT
オープン@1000	100	400					0		0
ページリード@1000	100	400	100 オープン				1	オープン ページリード 領域0	3ϕ
ページライト@1001	100	400	100 ページ リード				2	ページリード ページライト 領域0	2ϕ
オープン@1001	10	200	100 ページ ライト				3	オープン	0
ページリード@1001	100	400	100 ページ リード			200 オープン	0	ページライト ページリード 領域0	1

Fig. 24